NASA-CR-201092

MULTIDIMENSIONAL TRELLIS CODED PHASE MODULATION USING A MULTILEVEL CONCATENATION APPROACH

Technical Report

to

NASA

Electrical Engineering Division
Goddard Space Flight Center
Greenbelt, Maryland 20771

Grant Number NAG 5-931

Principal Investigator: Shu Lin

Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822

[illegible]y 24, 1996

MULTI-DIMENSIONAL TRELLIS CODED PHASE MODULATION USING A MULTILEVEL CONCATENATION APPROACH

Sandeep Rajpal, Do Jun Rhee and Shu Lin

Technical Report Number 96-003

May 24, 1996

Part I : CODE DESIGN

Abstract

The first part of this paper presents a simple and systematic technique for constructing multi-dimensional MPSK TCM codes. The construction is based on a multilevel concatenation approach, in which binary convolutional codes with good free branch distances are used as the outer codes and block MPSK modulation codes are used as the inner codes (or the signal spaces). Conditions on phase invariance of these codes are derived and a multi-stage decoding scheme for these codes is proposed. The proposed technique can be used to construct good codes for both the AWGN and fading channels as is shown in the second part of this paper.

1. Introduction

Since the publication of the celebrated paper by Ungerboeck in 1982 [1] on trellis coded modulation(TCM), there has been a boom of research in this area. Over the last fourteen years, researchers have proposed various techniques of constructing modulation codes using both convolutional codes (Trellis Coded Modulation (TCM))[1-7] and block codes (Block Coded Modulation (BCM))[8-14]. Almost all existing techniques for constructing TCM codes rely heavily on computer searches to find good TCM codes. These techniques work very well for small code complexities and rates. However, for large code complexities and high rates, the search becomes extremely time consuming (if not impossible) and a more systematic technique of construction is required. Most of the problems associated with algebraic construction of TCM codes arise due to the lack of indepth knowledge of convolutional codes. In addition, the nonlinearity of the mapping function (true for most signal constellations) which maps the coded output bits of the convolutional encoder onto the signal set, complicates the problem further. BCM codes on the other hand, have the advantage of being extremely rich in algebraic structure and phase symmetry, as has been shown in [10-13]. BCM codes however, have the disadvantage of being slightly poor in performance for low SNR (signal-to-noise ratio), as compared to TCM codes of the same decoding complexity, due to the large number of nearest neighbors.

Pietrobon et. al. extended Ungerboeck's results to multi-dimensional MPSK signal constellations [3]. They proposed a set partitioning technique for multi-dimensional MPSK signal constellations similar to Ungerboeck's set partitioning technique and then used computer search to design multi-dimensional MPSK TCM codes. However, due to the limitations of computer search, as were outlined above, they restricted themselves to 4 × 2-dimensions. In addition, to reduce the search complexity, they placed some other restrictions on the computer search. Multi-dimensional MPSK TCM codes have various advantages over 2-dimensional Ungerboeck TCM codes, the main ones being: (1) higher spectral efficiencies can be achieved; (2) codes constructed over multi-dimensional MPSK signal constellations have better phase invariance properties than that of 2-dimensional Ungerboeck MPSK codes; and (3) lower average decoding complexities to achieve the same performance.

A common point to be noted among all the construction techniques available in literature (whether TCM or BCM) is that the modulation codes constructed by these techniques require large decoding complexity to achieve large coding gains. The large decoding complexity of these codes makes them impractical for applications where high reliability and high data rates are required. As such, what is required is a **multi-stage** decoding technique which reduces the decoding complexity, while maintaining good performance.

This paper presents a simple and systematic technique for designing multi-dimensional

MPSK TCM codes with minimal computer search. The technique will be used to construct good codes for both the AWGN and fading channels. Though the main emphasis has been to construct codes for the MPSK signal constellation, the results are applicable to other signal constellations as well and modifying the existing construction for other signal constellations is straight forward. This paper is organized as follows: section 2 of the paper presents a new concept, branch distance of convolutional codes, which will be used extensively in the later sections. Section 3 outlines the basic construction technique of the proposed codes, and in addition shows that the codes constructed in [3] turn out to be a special case of the proposed construction. Section 4 discusses phase invariance. In section 5, a multi-stage decoding algorithm for the proposed codes is presented and it's decoding complexity is discussed. Section 6 concludes by discussing the design rules for constructing good codes using the proposed technique.

2. Branch Distance of Convolutional Codes

For two code sequences u and v in a binary linear convolutional code, the **branch distance** between them, denoted $d_b(\mathrm{u},\mathrm{v})$, is defined as the number of branches in which u and v differ (or equivalently, this is simply equal to the number of non-zero branches in $\mathrm{u}\oplus\mathrm{v}$, where $\oplus$ denotes binary addition). For a code sequence u in a binary linear convolutional code, the **branch weight** of u denoted $w_b(\mathrm{u})$ is simply the number of non-zero branches in u (or equivalently $w_b(\mathrm{u})$ is the **branch distance** between u and 0, where 0 refers to the all-zero code sequence, i.e., $w_b(\mathrm{u}) = d_b(\mathrm{u},\mathbf{0})$). The minimum free branch distance of a convolutional code C, denoted $d_{\text{B-free}}$, is the minimum branch distance between any two code sequences, i.e.,

$$d_{\text{B-free}} \triangleq \min\{d_b(\mathrm{u}\ ,\ \mathrm{v})\ :\ \mathrm{u},\mathrm{v} \in C \text{ and } \mathrm{u} \neq \mathrm{v}\} \tag{2.1}$$

Theorem 1: For a rate k/n feedforward binary linear convolutional code of total encoder memory γ, its minimum free branch distance, $d_{\text{B-free}}$, is upper bounded by $1 + \lfloor \gamma/k \rfloor$.

Proof: Let the k inputs to the encoder be denoted as $I_1, I_2, \cdots I_k$ and let the encoder memories associated with input I_i be γ_i for $1 \leq i \leq k$. Let $\min\{w_b(\mathrm{u})\}$ denote the minimum branch weight among all the code sequences associated with the binary linear convolutional code. Let $\min_{i=1}^{k} \gamma_i = \gamma_j$. Consider that the binary sequence $(1, 0, 0, \cdots)$ is fed into the input I_j and the all zero sequence $(0, 0, 0, \cdots)$ is fed into the remaining inputs. The branch weight of the resulting code sequence is upper bounded by $1 + \gamma_j$. Hence, $\min\{w_b(\underline{\mathrm{u}})\} \leq 1 + \gamma_j$. Since the code is linear, this also corresponds to an upper bound on the minimum free branch branch distance, i.e., $d_{\text{B-free}} \leq \{1 + \min_{i=1}^{k} \gamma_i\}$. Given any γ and k, the idea is to maximize $d_{\text{B-free}}$. Hence, $\max_{\gamma,k}(d_{\text{B-free}}) \leq \max_{\gamma,k}\{1 + \min_{i=1}^{k} \gamma_i\}$, i.e., the best $d_{\text{B-free}}$ for a given γ and k is $\leq \{1 + \max_{\gamma,k}\{\min_{i=1}^{k} \gamma_i\}\}$. It is readily seen that the value of $\max_{\gamma,k}\{\min_{i=1}^{k} \gamma_i\}$ is $\lfloor \gamma/k \rfloor$. $\Delta\Delta$

Theorem 2: If $d_{\text{B-free}} = 1 + \lfloor \gamma/k \rfloor$, then $N_{\text{B-free}}$, the number of codewords with branch weight $d_{\text{B-free}}$, is lower bounded by $(2^p - 1)$ where p is the number of inputs of the convolutional encoder which have an encoder memory of $\lfloor \gamma/k \rfloor$ associated with it.

Proof: Let $\mathbf{e_1}$ denote the binary sequence $(1, 0, 0, \cdots)$ i.e., 1 followed by the all zero sequence and let $\mathbf{e_0}$ denote the all zero binary sequence $(0, 0, 0, \cdots)$. Consider any non-zero code sequence $\mathbf{u}$. Then $w_b(\mathrm{u}) \geq 1 + \lfloor \gamma/k \rfloor$. Let the p inputs which have an encoder memory of $\lfloor \gamma/k \rfloor$ associated with it be I_j for $1 \leq j \leq p$. Consider that $\mathbf{e_0}$ is fed into the inputs I_j for $p + 1 \leq j \leq k$. Also, consider that the inputs I_j for $1 \leq j \leq p$ can take only one of the two sequences $\mathbf{e_0}$ or $\mathbf{e_1}$. Then the convolutional encoder under this constraint has $(2^p - 1)$ distinct non-zero input sequences. Each of the $(2^p - 1)$ sequences will have branch weight $\leq 1 + \lfloor \gamma/k \rfloor$. Since $d_{\text{B-free}} = 1 + \lfloor \gamma/k \rfloor$, each of the $(2^p - 1)$ sequences thus has branch weight $1 + \lfloor \gamma/k \rfloor$. Hence, $N_{\text{B-free}} \geq (2^p - 1)$. $\Delta\Delta$

A binary linear feed-forward convolutional code is said to be **optimal** in terms of branch distance if it achieves the upper bound as stated in theorem 1 for a given γ and k. Also, a code is said to be optimal in terms of the free Hamming distance, $d_{\text{H-free}}$, if it achieves the maximum $d_{\text{H-free}}$ possible for a given γ, k and n as specified in [15]. Note, from theorem 1, for a given $d_{\text{B-free}}$, higher encoder memory is required to achieve the same $d_{\text{B-free}}$ as k increases, i.e., given a certain fixed $d_{\text{B-free}}$, there is a tradeoff between complexity and rate. In addition, as is shown in theorem 2, $N_{\text{B-free}}$ also increases as the rate increases and hence there is also a tradeoff between rate and performance. A search has been performed on rate-1/2, -2/3 and -3/4 codes to find the best ones in terms of $d_{\text{B-free}}$ and $N_{\text{B-free}}$. The results are given in Tables 1 , 2 and 3.

An important point to note is that codes optimum in terms of branch distance may not be optimum in terms of the free Hamming distance $d_{\text{H-free}}$ and vice-versa. For small values of γ, it has been observed that codes optimum in terms of branch distance are also optimum in terms of $d_{\text{H-free}}$, however, the same does not hold for higher values of γ. From Table 1 we notice that up to $\gamma = 7$, the search yields codes which meet the upper bound in terms of $d_{\text{B-free}}$, however from that point on, the best codes start falling short of the upper bound by 1. Codes shown in Tables 2 and 3 meet the upper bound, however as the complexity increases, $N_{\text{B-free}}$ also starts increasing. Also listed in the tables is the $d_{\text{H-free}}$ and $N_{\text{H-free}}$, the number of codewords with $d_{\text{H-free}}$. The code generators in the tables have been listed in octal with the lowest degree on the left and the highest on the right, e.g., $(622)_8 \equiv 1 + D + D^4 + D^7$. As an example, consider the 8th code listed in Table 1. This is a rate-1/2 convolutional code with generators $1+D+D^4+D^7$ and $1+D^2+D^3+D^4+D^5+D^6+D^8$ and $d_{\text{B-free}} = 8$.

3. Construction of Multi-dimensional MPSK Codes

The proposed multi-dimensional MPSK codes are constructed using a q level concatenation approach as shown in Figure 1. Outer codes in the multi-level concatenation may be either block or convolutional, binary or non-binary. However, in this paper we will focus on binary convolutional codes as the outer codes.

Outer Codes:

The outer code, C_i, at the i-th level for $1 \leq i \leq q$ is chosen to be a convolutional code of rate k_i/n_i with optimum branch distance for the given rate and state-complexity. The parameters k_i and n_i depend upon the choice of the inner codes, as will be clear after the discussion of inner codes. Each outer code is selected from the tables mentioned in section 2. The reasons for selecting an optimum branch distance convolutional code will be clear when discussing theorems 4, 5 and 6.

Inner Codes:

Let S denote the two-dimensional MPSK signal constellation which ~~consists of~~ 2^{ℓ} signal points. Let S^m denote the set of all m-tuples over S, where m is a positive integer. Since S is a two-dimensional signal space, S^m is an $m \times 2$-dimensional signal space in which each signal point is a sequence of m MPSK signals. To construct the proposed codes, the signal space is chosen as a subspace of S^m, denoted Λ_0. In this paper, Λ_0 is constructed using the multilevel coding method proposed by Imai & Hirakawa [8].

Using the set partitioning approach proposed by Ungerboeck in [1], each signal point in the set S is labeled by a string of symbols from GF(2). Since S contains 2^{ℓ} signal points, we shall consider a labeling whose set of label strings is of the following form: $L \triangleq \{a_1 a_2 \cdots a_\ell : a_i \in \text{GF}(2) \text{ for } 1 \leq i \leq \ell\}$. Let λ denote the one-to-one mapping from L to S. If $a_1 a_2 \cdots a_\ell$ is the label for a signal point s, then $s = \lambda(a_1 a_2 \cdots a_\ell)$. Define an addition "+" on the label set L as follows: For two labels, $a_1 a_2 \cdots a_\ell$ and $a_1' a_2' \cdots a_\ell'$, in L, $a_1 a_2 \cdots a_\ell + a_1' a_2' \cdots a_\ell' = a_1'' a_2'' \cdots a_\ell''$ where $a_i'' = a_i \oplus a_i'$ for $1 \leq i \leq \ell$ and $\oplus$ is the modulo-2 addition. With this addition, L is simply the vector space of all ℓ-tuples over GF(2). We call L the label space for S.

For $1 \leq i \leq \ell$, let $C_{0,i}$ be a binary $(m, k_{0,i}, \delta_{0,i})$ linear block code of length m, dimension $k_{0,i}$ and minimum Hamming distance $\delta_{0,i}$. Let

$$\mathbf{V_i} = (v_{i,1}, v_{i,2}, \cdots v_{i,m}) \tag{3.1}$$

be a code word in $C_{0,i}$ for $1 \leq i \leq \ell$. We form the following sequence :

$$\mathbf{V_1} * \mathbf{V_2} * \cdots * \mathbf{V_\ell} \triangleq (v_{1,1}v_{2,1}\cdots v_{\ell,1} \text{ , } v_{1,2}v_{2,2}\cdots v_{\ell,2} \text{ , } \cdots, \ v_{1,m}v_{2,m}\cdots v_{\ell,m}) \tag{3.2}$$

For $1 \leq j \leq m$, we regard $v_{1,j}v_{2,j}\cdots v_{\ell,j}$ as the label for a signal point s_j in the MPSK signal set S. Then $\mathbf{V}_1 * \mathbf{V}_2 \cdots * \mathbf{V}_\ell$ is simply an m-tuple over the label set L and

$$\begin{aligned}\lambda(\mathbf{V}_1 * \mathbf{V}_2 \cdots * \mathbf{V}_\ell) &= (\lambda(v_{1,1}v_{2,1}\cdots v_{\ell,1}), \lambda(v_{1,2}v_{2,2}\cdots v_{\ell,2}), \cdots, \lambda(v_{1,m}v_{2,m}\cdots v_{\ell,m})) \\ &= (s_1, s_2, \cdots, s_m) \qquad (3.3)\end{aligned}$$

is an m-tuple over the MPSK signal set S (a sequence of m MPSK signals) which is a signal point in the $m \times 2$-dimensional signal space S^m. From codes $C_{0,i}$ for $1 \leq i \leq \ell$, we form the following set of m-tuples over the label set L :

$$C_{0,1} * C_{0,2} * \cdots * C_{0,\ell} = \{\mathbf{V}_1 * \mathbf{V}_2 * \cdots * \mathbf{V}_\ell : \mathbf{V}_1 \in C_{0,1}, \mathbf{V}_2 \in C_{0,2}, \cdots \mathbf{V}_\ell \in C_{0,\ell}\} \quad (3.4)$$

We will denote $C_{0,1} * C_{0,2} * \cdots * C_{0,\ell}$ by Ω_0. Then, Ω_0 is a vector space (or a linear code) over L (a subspace of the vector space of all the m tuples over L, denoted L^m). Ω_0 has $2^{k_{0,1}+k_{0,2}+\cdots+k_{0,\ell}}$ vectors. Hence, the dimension of Ω_0 is $\sigma_0 = k_{0,1} + k_{0,2} + \cdots + k_{0,\ell}$. Recall, that for $1 \leq i \leq q$, n_i denotes the number of output coded bits of the convolutional encoder at the i-th stage of encoding. Choose

$$n_1 + n_2 + \cdots + n_q = k_{0,1} + k_{0,2} + \cdots + k_{0,\ell} = \sigma_0 \qquad (3.5)$$

Suppose each m-tuple in Ω_0 is mapped into an m-tuple over the MPSK signal set S by the mapping $\lambda(\cdot)$. Then, we obtain the following subset of signal points in S^m:

$$\Lambda_0 \triangleq \lambda(\Omega_0) = \{\lambda(\mathbf{V}_1 * \mathbf{V}_2 \cdots * \mathbf{V}_\ell) : \mathbf{V}_1 \in C_{0,1}, \mathbf{V}_2 \in C_{0,2}, \cdots \mathbf{V}_\ell \in C_{0,\ell}\}$$

The set Λ_0 is a subspace of S^m with dimension σ_0. This subspace Λ_0 is actually a basic ℓ-level block MPSK modulation code of length m[8-14].

The performance of Λ_0 over the AWGN channel depends upon the minimum squared Euclidean distance and the number of nearest neighbors. The minimum squared Euclidean distance of Λ_0 can be calculated using results of [12]. On the other hand, the performance of Λ_0 over fading channels depends upon the minimum symbol distance, product distance, number of nearest neighbors and the squared Euclidean distance to a lesser extent [17]. The minimum symbol distance of Λ_0 is given by [17] $\delta_H^0 = \min_{i=1}^{\ell} \delta_{0,i}$. Suppose, Λ_0 has minimum squared Euclidean distance Δ_0^2 and minimum symbol distance δ_H^0.

In the following, the subspace Λ_0 of S^m will be used as the signal space for constructing multi-dimensional trellis MPSK codes. Before presenting the code construction, we need to define a subspace of Ω_0 for partitioning Ω_0. For $1 \leq j \leq \ell$, let $C_{1,j}$, $C_{2,j}$, $\cdots$, $C_{q,j}$ be a sequence of linear subcodes of $C_{0,j}$ such that

$$C_{q,j} \subseteq C_{q-1,j} \subseteq \cdots \subseteq C_{1,j} \subseteq C_{0,j}. \tag{3.6}$$

Let $k_{i,j}$ be the dimension and $\delta_{i,j}$ be the minimum Hamming distance of $C_{i,j}$ for $1 \leq i \leq q$. Then $C_{i,j}$ is an $(m,\ k_{i,j},\ \delta_{i,j})$ code. For $1 \leq i \leq q$, we form the following linear code over the labeling space L: $\Omega_i = C_{i,1} * C_{i,2} * \cdots * C_{i,\ell}$. The dimension of this code is $\sigma_i = k_{i,1} + k_{i,2} + \cdots + k_{i,\ell}$. It is clear that for $1 \leq i \leq q$,

$$\Omega_i \subseteq \Omega_{i-1} \tag{3.7}$$

It follows from (3.7) that Ω_1, Ω_2, $\cdots$, Ω_q form a sequence of subspaces of Ω_0 and

$$\Omega_q \subseteq \Omega_{q-1} \subseteq \cdots \subseteq \Omega_1 \subseteq \Omega_0 \tag{3.8}$$

For $1 \leq i \leq q$, let

$$\Lambda_i \triangleq \lambda(\Omega_i) \tag{3.9}$$

Then, Λ_i is a subspace of S^m with dimension $\dim(\Lambda_i) = \sigma_i$. Let the minimum squared Euclidean distance of Λ_i be Δ_i^2 and minimum symbol distance be δ_H^i. Equations (3.8) and (3.9) imply that Λ_1, Λ_2, $\cdots$, Λ_q form a sequence of subspaces of Λ_0 and

$$\Lambda_q \subseteq \Lambda_{q-1} \subseteq \cdots \subseteq \Lambda_1 \subseteq \Lambda_0 \tag{3.10}$$

Suppose the binary codes, $C_{i,j}$ with $1 \leq i \leq q$ and $1 \leq j \leq \ell$, are chosen such that

$$n_i = \sigma_{i-1} - \sigma_i \tag{3.11}$$

It follows from (3.5) and (3.11) that

$$\begin{aligned}
\sigma_1 &= n_2 + n_3 + \cdots + n_q \\
\sigma_2 &= n_3 + \cdots + n_q \\
&\vdots \\
\sigma_{q-1} &= n_q \\
\sigma_q &= 0
\end{aligned}$$

Ω_0 and its subcodes $\Omega_1, \Omega_2, \cdots, \Omega_q$ are used to form a sequence of coset codes [7]. Let $\mathbf{U_1} * \mathbf{U_2} * \cdots * \mathbf{U}_\ell$ be a vector in Ω_0 but not in Ω_1. Then $\mathbf{U_1} * \mathbf{U_2} \cdots * \mathbf{U}_\ell + \Omega_1$ is a coset of Ω_1 in Ω_0 and $\mathbf{U_1} * \mathbf{U_2} * \cdots * \mathbf{U}_\ell$ is called the **coset representative**. Recall $n_1 = \sum_{i=1}^{\ell}(k_{0,i} - k_{1,i})$. Hence, there are 2^{n_1} cosets of Ω_1 in Ω_0. These 2^{n_1} cosets of Ω_1 form a partition of Ω_0. Let Ω_0/Ω_1 denote the set of cosets in Ω_0 modulo Ω_1. Ω_0/Ω_1 is called a coset code. Let $[\Omega_0/\Omega_1]$ denote the set of coset representatives of the coset code Ω_0/Ω_1. Hence $\Omega_0/\Omega_1 = [\Omega_0/\Omega_1] + \Omega_1$. Ω_1 can be further partitioned using Ω_2, in the same way as is outlined above. Partitioning each coset of Ω_1 in Ω_0 on the basis of Ω_2, we form the coset code $\Omega_0/\Omega_1/\Omega_2$. Let $[\Omega_1/\Omega_2]$ denote the set of coset representatives in the partition Ω_1/Ω_2. Hence each coset in the coset code $\Omega_0/\Omega_1/\Omega_2$ can be written in the form $[\Omega_0/\Omega_1] + [\Omega_1/\Omega_2] + \Omega_2$. Proceeding in this manner, we form the following sequence of coset codes:

$$\begin{aligned} B_1 &= \Omega_0/\Omega_1 \\ B_2 &= \Omega_0/\Omega_1/\Omega_2 \\ &\vdots \\ B_q &= \Omega_0/\Omega_1/\Omega_2/\cdots/\Omega_q \end{aligned}$$

For $1 \leq i \leq q$, each coset in $B_{i-1} = \Omega_0/\Omega_1/\cdots/\Omega_{i-1}$ consists of 2^{n_i} cosets modulo Ω_i. These coset codes are used as the inner codes in the multi-level concatenation in which B_1 is used at the first level and B_q at the q-th level.

Let ω_0 and ω_0' be two distinct points in Ω_0. If these two points are in two distinct cosets of B_1 then the squared Euclidean distance between $s = \lambda(\omega_0)$ and $s' = \lambda(\omega_0')$ is at least Δ_0^2. If the two points ω_0 and ω_0' are in the same coset of B_1 but distinct cosets of B_2, then the squared Euclidean distance between s and s' is at least Δ_1^2. Generalizing in this manner, it is easy to see that if the two points ω_0 and ω_0' have identical coset representatives in B_j for $1 \leq j < i$, but distinct coset representatives for B_i then s and s' have a squared Euclidean distance of at least Δ_{i-1}^2. Hence, B_1 is the least powerful and B_q is the most powerful coset code in terms of Euclidean distance.

The same arguments as above will also hold if the minimum squared Euclidean distance at each stage is replaced by the corresponding minimum symbol distance.

Encoding of the $m \times 2$-dimensional TCM code:

Encoding is accomplished in q stages, as shown in Figure 1, and for $1 \leq i \leq q$, the i-th level encoding is accomplished in two steps: (1) at any time instant t, a message of k_i bits is encoded based on the convolutional outer code C_i into an n_i-bit coded block; and (2) the n_i-bit code block then selects a coset from the coset code $B_i = \Omega_0/\Omega_1/\cdots/\Omega_{i\cdot}$ _

The output at the i-th level encoder is a sequence of cosets from B_i. All the possible coset sequences at the i-th level form a trellis, and each branch in the trellis corresponds to a coset in B_i, and this trellis is isomorphic to the trellis of C_i. Let $\mathrm{v_i}$ denote a code sequence in the convolutional code C_i and let ϕ_i denote the mapping from the n_i coded output bits of the convolutional code to the 2^{n_i} cosets. Hence, $\phi_i(\mathrm{v_i})$ denotes the sequence of coset representatives at the i-th stage of encoding, corresponding to the code sequence $\mathrm{v_i}$. Hence, any code sequence in the $m \times 2$-dimensional TCM code can be written in the form

$$\lambda(\phi_1(\mathrm{v_1}) + \phi_2(\mathrm{v_2}) + \cdots + \phi_q(\mathrm{v_q})). \tag{3.12}$$

At every time instant t, the encoder puts out m MPSK signals.

A very interesting and special case of the proposed codes occurs when $q = 2$ and the second level outer code is left uncoded, as shown in Figure 2. This structure is equivalent to the structure used for the construction of the multi-dimensional codes in [3]. A computer search was used in [3] to find the convolutional code to be used at the first level. The computer search selected a convolutional code which optimized the multi-dimensional code both in terms of Euclidean distance and number of nearest neighbors.

A multi-dimensional code is said to be **linear** with respect to binary addition, if for any two code sequences in the multi-dimensional code, $\mathbf{U} = \lambda(\phi_1(\mathrm{u_1}) + \phi_2(\mathrm{u_2}) + \cdots + \phi_q(\mathrm{u_q}))$ and $\mathbf{V} = \lambda(\phi_1(\mathrm{v_1}) + \phi_2(\mathrm{v_2}) + \cdots + \phi_q(\mathrm{v_q}))$,

$$\mathbf{U} \oplus \mathbf{V} \triangleq \lambda((\phi_1(\mathrm{u_1}) + \phi_2(\mathrm{u_2}) + \cdots + \phi_q(\mathrm{u_q})) + (\phi_1(\mathrm{v_1}) + \phi_2(\mathrm{v_2}) + \cdots + \phi_q(\mathrm{v_q})))$$

is also a code sequence, where $\mathrm{u_i}$ and $\mathrm{v_i}$ for $1 \leq i \leq q$ denote output code sequences of the convolutional code encoder C_i at the i-th level. Linearity of the code (in terms of binary addition) simplifies the error analysis and in addition leads to a simpler encoder and decoder. The linear structure leads to the following theorems on the linearity, minimum squared Euclidean distance and minimum symbol distance of the proposed codes.

Theorem 3 : A multi-dimensional code is linear with respect to binary addition, if all the mappings ϕ_i, for $1 \leq i \leq q$ are linear.

Proof : Recall, that any code sequence in a multi-dimensional TCM code can be written in the form $\lambda(\phi_1(\mathbf{v_1}) + \phi_2(\mathbf{v_2}) + \cdots + \phi_q(\mathbf{v_q}))$ where $\mathbf{v_i}$ for $1 \leq i \leq q$ denotes the output code sequence of the convolutional code C_i at the i-th level. The proof then follows trivially from the definition of linearity.

Theorem 4 : The minimum free squared Euclidean distance of a coset trellis code at the j-th level, for $1 \leq j \leq q$ is lower bounded by $D^2_{(j),\text{free}} \geq \Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}$, where $d^{(j)}_{\text{B-free}}$ denotes the minimum free branch distance of the convolutional code at the j-th level, C_j.

Proof : Consider two distinct code sequences, $\mathbf{U} = \lambda(\phi_1(\mathbf{u_1}) + \phi_2(\mathbf{u_2}) + \cdots + \phi_q(\mathbf{u_q}))$ and $\mathbf{V} = \lambda(\phi_1(\mathbf{v_1}) + \phi_2(\mathbf{v_2}) + \cdots + \phi_q(\mathbf{v_q}))$, where $\mathbf{u_i}$ and $\mathbf{v_i}$ for $1 \leq i \leq q$ denotes two output code sequences of the convolutional code C_i at the i-th level. Assume that $\mathbf{u_i} = \mathbf{v_i}$ for $1 \leq i < j$ and $\mathbf{u_j} \neq \mathbf{v_j}$. At a particular time instant t, let $\lambda(\omega)$ and $\lambda(\omega')$ be the corresponding transmitted signal points for $\mathbf{U}$ and $\mathbf{V}$ respectively, where ω and $\omega' \in \Omega_0$. Since $\mathbf{u_i} = \mathbf{v_i}$ for $1 \leq i < j$ and $\mathbf{u_j} \neq \mathbf{v_j}$, hence ω and ω' have identical coset representatives in B_i for $1 \leq i < j$ and hence the minimum squared Euclidean distance between $\lambda(\omega)$ and $\lambda(\omega')$ is at least Δ^2_{j-1}. Since C_j has minimum free branch distance $d^{(j)}_{\text{B-free}}$, hence the two sequences $\mathbf{u_j}$ and $\mathbf{v_j}$ are distinct in at least $d^{(j)}_{\text{B-free}}$ branches. Therefore, the squared Euclidean distance between $\mathbf{U}$ and $\mathbf{V}$ is at least $\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}$. $\Delta\Delta$

Theorem 5 : The minimum free squared Euclidean distance of the overall TCM code is lower bounded by $D^2_{\text{free}} \geq \min_{1 \leq j \leq q}\{\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}\}$.

Proof : Consider two distinct code sequences $\mathbf{U}$ and $\mathbf{V}$. Using the same notation as developed in theorem 4, consider that $\mathbf{u_i} = \mathbf{v_i}$ for $1 \leq i < j$ and that $\mathbf{u_j} \neq \mathbf{v_j}$. Then, theorem

4 gives us the minimum squared Euclidean distance between the two sequences. Since j is arbitrary, the minimum squared Euclidean distance between the two sequences is obtained by taking the minimum over all the q levels, i.e., if $D^2(\mathbf{U}, \mathbf{V})$ denotes the squared Euclidean distance between the two sequences $\mathbf{U}$ and $\mathbf{V}$, then $D^2(\mathbf{U}, \mathbf{V}) \geq \min_{1 \leq j \leq q}\{\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}\}$. Since $\mathbf{U}$ and $\mathbf{V}$ are any two sequences, the theorem follows. $\Delta\Delta$

Theorem 6 : The minimum symbol distance of the overall TCM code is lower bounded by $\delta_H \geq \min_{1 \leq j \leq q}\{\delta^j_H \cdot d^{(j)}_{\text{B-free}}\}$.

Proof : The proof is similar to that in theorem 5, with the only difference that instead of minimum squared Euclidean distance we now consider minimum symbol distance. $\Delta\Delta$

4.A Spectral Efficiency

At each encoding time instant, $k_1 + k_2 + \cdots + k_q$ bits are fed into the encoder (Figure 1), and the corresponding output is m MPSK signals. Hence the spectral efficiency of the $m \times 2$-dimensional TCM code is $(k_1 + k_2 + \cdots + k_q)/m$ bits/symbol.

4.B Phase Invariance

Phase symmetry of a code is important in resolving carrier-phase ambiguity and ensuring rapid carrier-phase resynchronization after temporary loss of synchronization [2]. It is desirable for a modulation code to have as many phase symmetries as possible. Recall, that the proposed multi-dimensional modulation codes are constructed using q convolutional codes and $q + 1$ basic ℓ-level block modulation codes (Figure 1). The phase invariance of the proposed codes is a function of both the inner codes and the outer codes. If convolutional codes are used at all the q levels, the phase invariance of the constructed modulation codes would depend upon the structure of the convolutional codes used, and for most cases the constructed modulation codes would have no phase invariance. A special case of the proposed codes occurs when the outer code at the q-th level is left uncoded (figure 2 shows this special case for $q = 2$). Most of the codes constructed using this special case do have phase invariance. Kasami et. al. in [16] derived conditions on phase invariance of basic

ℓ-level block modulation codes. A slightly modified form of the conditions proposed in [16] will be applicable to the proposed codes.

The following theorem gives the conditions for the proposed modulation codes to be phase invariant under rotation for this special case.

Theorem 7: Let $\Lambda_0 = \lambda(C_{0,1} * C_{0,2} * \cdots * C_{0,\ell})$ and let $\Lambda_{q-1} = \lambda(C_{q-1,1} * C_{q-1,2} * \cdots * C_{q-1,\ell})$, where $C_{0,i}$ and $C_{q-1,i}$ for $1 \le i \le \ell$ are binary linear block codes of length m. For $1 \le h \le \ell$, the multi-dimensional MPSK TCM code is phase invariant under $180°/2^{\ell-h}$ phase shifts if the multi-dimensional TCM code is linear with respect to binary addition and:

$$\mathbf{1} \in C_{q-1,h} \text{ and} \tag{4.1}$$

$$C_{0,h} \cdot C_{0,h+1} \cdots C_{0,j-1} \subseteq C_{q-1,j} \text{ for } h < j \le \ell \tag{4.2}$$

where $\mathbf{1}$ denotes the all-one binary sequence of length m, and for two-binary m-tuples $\mathbf{a} = (a_1, a_2, \cdots a_m)$ and $\mathbf{b} = (b_1, b_2, \cdots b_m)$, $\mathbf{a} \cdot \mathbf{b} \triangleq (a_1 \cdot b_1, a_2 \cdot b_2, \cdots, a_m \cdot b_m)$, where $a_i \cdot b_i$, for $1 \le i \le m$ denotes the logical product of a_i and b_i.

Proof: Appendix A

If the outer code at the q-th level is left uncoded, sequences of signal points from Λ_{q-1} are valid code sequences. The best phase invariance that can be achieved for the overall multi-dimensional code in this case is equal to the phase invariance of Λ_{q-1}. The conditions as stated in theorem 7 provide a set of conditions which guarantee a certain phase invariance for the overall multi-dimensional MPSK TCM code independent of the convolutional codes chosen. Most codes designed using the proposed technique, do achieve the best possible phase invariance (i.e., of Λ_{q-1}).

5. Multi-stage Decoding Algorithm

One obvious way of decoding a TCM code proposed in section 3, is to form a super trellis for the code, which is obtained by taking the direct product of the trellises of the convolutional codes at the q levels. The complexity associated with this technique (for most cases) would be tremendous. We will focus on a multi-stage decoding scheme, in which the deooding is carried out in q stages, corresponding to the q levels of the multi-dimensional TCM code. Let $\mathbf{V} = (s_1, s_2, s_3, \cdots)$ be the transmitted code sequence, where s_i for $1 \leq i \leq \infty$ denotes a signal point in the MPSK signal constellation and let $\mathbf{R} = (r_1, r_2, r_3, \cdots)$ denote the corresponding received sequence. Using (3.12), $\mathbf{V}$ can be written in the form $\mathbf{V} = \lambda(\phi_1(\mathbf{v}_1) + \phi_2(\mathbf{v}_2) + \cdots + \phi_q(\mathbf{v}_q))$ where $\mathbf{v}_\mathbf{i}$ for $1 \leq i \leq q$ denotes a code sequence in the convolutional code C_i.

First stage of decoding:

At the first stage, $\mathbf{v}_1$ is estimated using the received sequence $\mathbf{R}$. Recall, that at the first stage of encoding, the trellis is isomorphic to the trellis of the convolutional code C_1 used at the first level, with each branch of the trellis corresponding to a coset in B_1. Each coset in B_1 can be written in the general form $\omega_0 + \Omega_1$, where $\omega_0 \in [\Omega_0/\Omega_1]$. Let us call this isomorphic trellis $\tilde{C}_1$. Hence, each branch of $\tilde{C}_1$ consists of 2^{σ_1} points, corresponding to the 2^{σ_1} points in Ω_1. The trellis $\tilde{C}_1$ is used to form the trellis $\lambda(\tilde{C}_1)$, where

$$\lambda(\tilde{C}_1) \triangleq \{\lambda(\mathbf{v}) : \mathbf{v} \in \tilde{C}_1\}. \tag{5.1}$$

The trellis $\lambda(\tilde{C}_1)$ will be used for decoding at the first stage. Any code sequence in $\lambda(\tilde{C}_1)$ can be written in the form

$$\lambda(\phi_1(\mathbf{u}_1) + \omega_1) \tag{5.2}$$

where $\mathbf{u}_1$ is a code sequence in C_1 and ω_1 is a sequence of points from Ω_1, i.e., $\omega_1 = \{(\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \cdots) : \omega_{1,i} \in \Omega_1 \text{ for } 1 \leq i \leq \infty\}$. Standard soft-decision Viterbi decoding[2] is performed on $\mathbf{R}$ using the trellis $\lambda(\tilde{C}_1)$. This yields a code sequence $\lambda(\phi_1(\hat{\mathbf{v}}_1) + \hat{\omega}_1)$ in $\lambda(\tilde{C}_1)$ which is closest to the received sequence $\mathbf{R}$ in terms of squared Euclidean distance. The code sequence $\hat{\mathbf{v}}_1$ forms an estimate of the sequence $\mathbf{v}_1$. $\hat{\omega}_1$ denotes a sequence of points from Ω_1. Since $\hat{\mathbf{v}}_1$ is a code sequence in C_1, the estimate of the information sequence associated with the first level can be obtained from $\hat{\mathbf{v}}_1$.

[2] We will use minimum squared Euclidean distance as the decoding metric for both the AWGN and fading channels.

The i-th stage of decoding:

The second and subsequent stages of decoding are very similar to the first stage of decoding. For $2 \leq i \leq q$, let us consider the i-th stage of decoding. The previous $i-1$ stages of decoding give us estimates of $\mathbf{v_j}$, denoted by $\hat{\mathbf{v}}_j$ for $1 \leq j \leq (i-1)$. Using arguments similar to that given above, we form the **isomorphic** trellis $\tilde{C}_i$, where any code_sequence in $\tilde{C}_i$ can be expressed in the general form

$$\phi_1(\hat{\mathbf{v}}_1) + \phi_2(\hat{\mathbf{v}}_2) + \cdots + \phi_{i-1}(\hat{\mathbf{v}}_{i-1}) + \phi_i(\mathbf{u_i}) + \omega_i \tag{5.3}$$

where $\mathbf{u_i}$ is a code sequence in the convolutional code at the i-th level, C_i and ω_i is sequence of points from Ω_i. Each branch of $\tilde{C}_i$ consists of 2^{σ_i} points, corresponding to the number of points in Ω_i. The trellis $\tilde{C}_i$ is used to form the trellis $\lambda(\tilde{C}_i)$, where

$$\lambda(\tilde{C}_i) \triangleq \{\lambda(\mathbf{v}) : \mathbf{v} \in \tilde{C}_i\}. \tag{5.4}$$

The trellis $\lambda(\tilde{C}_i)$ will be used for decoding at the i-th stage. Standard soft-decision Viterbi decoding is performed on $\mathbf{R}$ using the trellis $\lambda(\tilde{C}_i)$. This yields a code sequence

$$\lambda(\phi_1(\hat{\mathbf{v}}_1) + \phi_2(\hat{\mathbf{v}}_2) + \cdots + \phi_{i-1}(\hat{\mathbf{v}}_{i-1}) + \phi_i(\hat{\mathbf{v}}_i) + \hat{\omega}_i) \tag{5.5}$$

in $\lambda(\tilde{C}_i)$ which is closest to the received sequence $\mathbf{R}$ in terms of squared Euclidean distance, where $\hat{\mathbf{v}}_i$ is a code sequence in the convolutional code used at the i-th level, C_i, and $\hat{\omega}_i$ is a sequence of points from Ω_i. The code sequence $\hat{\mathbf{v}}_i$ forms an estimate of the sequence $\mathbf{v_i}$. Since $\hat{\mathbf{v}}_i$ is a code sequence in C_i, the information sequence associated with the i-th level can be obtained from $\hat{\mathbf{v}}_i$.

The branch metric (squared Euclidean distance) for each branch in $\lambda(\tilde{C}_i)$, $1 \leq i \leq q$, is calculated by taking the m received signals corresponding to that branch and finding the element in the coset corresponding to that branch, which is closest to the m received signals in terms of Euclidean distance. This process of finding the closest element in the coset is termed as **closest coset decoding.** The Euclidean distance corresponding to the closest element in the coset becomes the branch metric. If m is small, calculation of the branch metric does not represent a formidable task, however if m is large and if Ω_i, $1 \leq i \leq q$, has trellis structure then a trellis can be used to calculate the branch metric. In addition, if the number of states associated with the trellis structure of Ω_i is big, multi-stage decoding for Ω_i can be used to further reduce the decoding complexity. Multi-stage decoding of Ω_i would be carried out in the same way as proposed in [10, 11].

Another way of reducing the decoding complexity associated with **closest coset decoding** would be as follows: Consider a trellis $\tilde{C}_i^{\text{sup}}$, where any code sequence in the trellis $\tilde{C}_i^{\text{sup}}$ can be written in the following form:

$$\phi_1(\hat{\mathrm{v}}_1) + \phi_2(\hat{\mathrm{v}}_2) + \cdots + \phi_{i-1}(\hat{\mathrm{v}}_{i-1}) + \phi_i(\mathrm{u_i}) + \omega_i^{\text{sup}} \tag{5.6}$$

where ω_i^{sup} is a sequence of points from Ω_i^{sup}, and the rest of the sequences are as before. If $\Omega_i \subset \Omega_i^{\text{sup}}$ then the trellis $\tilde{C}_i$ is a subcode of the trellis $\tilde{C}_i^{\text{sup}}$. As such, instead of using $\tilde{C}_i$ we can use $\tilde{C}_i^{\text{sup}}$ at the i-th stage of decoding. Ω_i^{sup} can be chosen to have a simpler trellis structure as compared to that of Ω_i. This would reduce the complexity associated with **closest coset decoding** and hence reduce the decoding complexity associated with the i-th stage of decoding.

Multi-stage decoding leads to error propagation. To reduce the effect of error propagation, the first couple of decoding stages should be powerful. A special case of the decoding algorithm occurs for $q = 2$ and $k_2 = n_2$. If **closest coset decoding** at the first stage is carried out in a single-stage, then the overall decoding of the multi-dimensional code is also one-stage. If m is small, then one-stage **closest coset decoding** is feasible, however if m is large, multi-stage **closest coset decoding** could be adopted to reduce the decoding complexity. The overall decoding in the latter case would then be multi-stage.

Decoding complexity of the proposed decoding algorithm :

The complexity of the proposed schemes will be measured in terms of the number of computations required for the decoder to produce an estimate of each 2-dimensional PSK signal. For $1 \leq i \leq q$, let γ_i be the total encoder memory of the convolutional code used at the i-th level in the proposed scheme. Consider the i-th stage of decoding. Then, due to the Viterbi algorithm alone, the complexity is $2^{\gamma_i+k_i}$ additions and $2^{\gamma_i}(2^{k_i}-1)$ comparisons, per $m \times 2$-dimensions (since each branch has m MPSK signals). The branch metric calculation forms an additional complexity and depends upon the choice of the inner codes. Let us call this complexity B_{C_i}. Hence the total complexity per $m \times 2$-dimensions is : (1) $\sum_{i=1}^{q} 2^{\gamma_i+k_i}$ additions; (2) $\sum_{i=1}^{q} 2^{\gamma_i}(2^{k_i}-1)$ comparisons; and $\sum_{i=1}^{q} B_{C_i}$. Dividing this total complexity by m would give us the number of computations required per 2-dimensions (i.e., the number of computations required to decode a single MPSK point).

6. Design Rules for Good Codes

The performance of codes designed using the proposed technique depends upon various factors. If all the design considerations are followed strictly, the codes usually would achieve good performance and in some cases, with reduced decoding complexity. Some of the most important design considerations are: (1) the number of levels q, in the multilevel concatenation should be kept as low as possible. The advantages of this are twofold. First, reducing the number of encoding levels, would reduce the number of decoding stages and in most cases reduce the decoding complexity. Secondly, reducing the number of decoding levels also decreases the amount of error propagation which occurs as a result of the multi-stage decoding. To reduce the error propagation due to multi-stage decoding, the first few levels should be chosen extremely powerful, so that the amount of error propagation is decreased. This however leads to higher decoding complexity for the first few levels; (2) the number of dimensions, i.e., $m \times 2$, should be kept as low as possible. As m increases the number of nearest neighbors associated with the code also start increasing, which limits the performance of the code. On the other hand, increasing m usually helps in decreasing the normalized decoding complexity associated with the code; (3) theorem 5 gives us the minimum squared Euclidean distance of the overall multi-dimensional TCM code. For a given minimum squared Euclidean distance of the TCM code, $d_{\text{B-free}}$ of the convolutional codes chosen to form the multi-dimensional TCM code should be chosen to be as small as possible. Lower $d_{\text{B-free}}$ would imply lower decoding complexity associated with the convolutional code decoding. The above also holds for theorem 6; (4) the branch computation complexity B_{C_i} at the i-th stage of decoding depends upon Λ_i. If Λ_i is chosen to have a simple trellis structure, the corresponding branch computation complexity will be minimal. If on the other hand, the trellis for Λ_i is sufficiently complex, techniques described in section 5 can be used to reduce the computation complexity. These techniques however, usually lead to degraded performance; (5) construction of codes with good phase invariance, places restrictions on codes as per theorem 7 and hence in most cases this would limit either the performance and/or the achievable spectral efficiency.

Most design considerations mentioned above lead to conflicting requirements. Hence, there is a tradeoff involved between performance, decoding complexity, spectral efficiency and phase invariance.

Appendix A

Proof of theorem 7: The proof follows very closely the derivation of the phase invariance conditions in [16]. For the code to be phase invariant by $180°/2^{\ell-h}$, any code sequence in the multi-dimensional code when rotated by $180°/2^{\ell-h}$ should produce another code sequence. Let $\mathbf{V}$ be the transmitted code sequence. Let $\mathbf{V}^{\text{rot}}$ denote the code sequence $\mathbf{V}$ rotated by $180°/2^{\ell-h}$. Recall from section 3, that the basic building block of the proposed multi-dimensional codes is Λ_0, hence any valid code sequence in the multi-dimensional code can be considered to be a sequence of points from Λ_0. Consider the j-th time instant. Let $\mathbf{V_j} = \lambda(\mathbf{V_{1j}} * \mathbf{V_{2j}} * \cdots * \mathbf{V_{\ell j}})$ be the transmitted sequence of m MPSK signals at the j-th time instant, where, $\mathbf{V_{ij}} \in C_{0,i}$ for $1 \le i \le \ell$. Also, let $\mathbf{V_j^{rot}} = \lambda(\mathbf{V_{1j}^{rot}} * \mathbf{V_{2j}^{rot}} * \cdots * \mathbf{V_{\ell j}^{rot}})$ be the sequence of m MPSK signals for $\mathbf{V}^{\text{rot}}$ at the j-th time instant, where, $\mathbf{V_{ij}^{rot}} \in C_{0,i}$ for $1 \le i \le \ell$. Using results of [16], $\mathbf{V_j^{rot}}$ can be written in the following form :

$$\mathbf{V_j^{rot}} = \lambda((\mathbf{V_{1j}} + \mathbf{V'_{1j}}) * (\mathbf{V_{2j}} + \mathbf{V'_{2j}}) * \cdots * (\mathbf{V_{\ell j}} + \mathbf{V'_{\ell j}})), \qquad (A.1)$$

where $\mathbf{V'_{ij}} = 0$ for $1 \le i < h$, $\mathbf{V'_{hj}} = \mathbf{1}$ and $\mathbf{V'_{ij}} = \mathbf{V_{hj}} \cdot \mathbf{V_{h+1j}} \cdots \mathbf{V_{i-1j}}$ for $h < i \le \ell$ and

$\mathbf{0}$ denotes the all-zero sequence of length m. Form the sequence $\mathbf{V}'$, such that the j-th time instant of $\mathbf{V}'$ is:

$$\mathbf{V}'_{\mathbf{j}} = \lambda(\mathbf{V}'_{\mathbf{1j}} * \mathbf{V}'_{\mathbf{2j}} * \cdots * \mathbf{V}'_{\ell\mathbf{j}}) \qquad (A.2)$$

Then, for the code to be phase invariant under rotations of $180°/2^{\ell-h}$, $\mathbf{V}'$ should also be a valid code sequence. Sequences of signal points from Λ_{q-1} form a valid code sequence. Hence, if $\mathbf{V}'_{\mathbf{j}} \in \Lambda_{q-1}$ then $\mathbf{V}$ is phase invariant under rotations of $180°/2^{\ell-h}$, i.e., if $\mathbf{1} \in C_{q-1,h}$ and

$$\mathbf{V}_{\mathbf{hj}} \cdot \mathbf{V}_{\mathbf{h+1j}} \cdots \mathbf{V}_{\mathbf{i-1j}} \in C_{q-1,i} \text{ for } h < i \leq \ell, \qquad (A.3)$$

then $\mathbf{V}$ is phase invariant under phase rotations of $180°/2^{\ell-h}$. Since the above should hold for any transmitted sequence $\mathbf{V}$, the theorem follows.

References

[1] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," **IEEE Trans. on Information Theory**, Vol. IT-28, No. 1, pp. 55-67, January 1982.

[2] G. Ungerboeck, "Trellis-coded Modulation with Redundant Signal Sets: Parts I and II," **IEEE Communication Magazine**, Vol. 25, pp. 5-21, Feb. 1987.

[3] S.S. Pietrobon et. al., "Trellis Coded Multi-Dimensional Phase Modulation," **IEEE Trans. on Information Theory**, Vol. IT - 36, No. 1 , pp. 63 - 89, January 1990.

[4] A.R. Calderbank and N.J.A. Sloane, "New Trellis Codes Based on Lattices and Cosets," **IEEE Trans. on Information Theory**, Vol. IT-33, pp. 177-195, March 1987.

[5] A. R. Calderbank and J. E. Mazo, "A New Description of Trellis Codes," **IEEE Trans. on Information Theory**, Vol. IT-30, pp. 784-791, Nov. 1984.

[6] L.F. Wei, "Trellis-Coded Modulation with Multi-dimensional Constellations," **IEEE Trans. on Information Theory**, Vol. IT-33, pp. 483-501, July 1987.

[7] G.D. Forney, Jr., "Coset Codes - Part I : Introduction and Geometrical Classification," **IEEE Trans. on Information Theory**, Vol. IT - 34, pp. 1123 - 1151, Sep. 1988.

[8] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes," **IEEE Trans. on Information Theory**, Vol. IT-23, pp. 371-376, May 1977.

[9] V.V. Ginzburg, "Multidimensional Signals for a Continuous Channel," **Problemy Peredachi Informatsii**, Vol. 20, No. 1, pp. 28-46, 1984.

[10] S.I. Sayegh, "A Class of Optimum Block Codes in Signal Space," **IEEE Trans. on Communications**, Vol. COM-30, No. 10, pp. 1043-1045, October 1986.

[11] R.M. Tanner, "Algebraic Construction of Large Euclidean Distance Combined Coding Modulation Systems," **Abstract of Papers, 1986 IEEE International Symposium on Information Theory**, Ann Harbor, October 6-9, 1986.

[12] T. Kasami et. al., "On Multi-Level Block Modulation Codes," **IEEE Trans. on Information Theory**, Vol. IT - 37, No. 4 , pp. 965 - 975, July 1991.

[13] A.R. Calderbank, "Multi-Level Codes and Multi-Stage Decoding," **IEEE Trans. on Communications**, Vol. COM-37, No. 3, pp. 222-229, March 1989.

[14] T. Kasami et. al., "A Concatenated Coded Modulation Scheme for Error Control," **IEEE Trans. on Communications**, Vol. COM-38, No. 6, pp. 752-763, June 1990.

[15] S. Lin and D.J. Costello, Jr., **Error Control Coding: Fundamentals and Applications**, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[16] T. Kasami et. al., "On Linear Structure and Phase Rotation Invariant Properties of Block 2^l - PSK Modulation Codes," **IEEE Trans. on Information Theory**, Vol. IT - 37, No. 1 , pp. 164 - 167, January 1991.

[17] D. Divsalar and M.K. Simon, "The Design of Trellis Coded MPSK for Fading Channels : Performance Criteria," **IEEE Trans. on Comm.**, Vol. 36, pp. 1004 - 1012, Sep. 1988.

Table 1 Optimum Branch Distance Rate 1/2 Codes

$\gamma^{\dagger}$	G	$d_{\text{B-free}}^{\ddagger}$	$N_{\text{B-free}}^{\triangle}$	$d_{\text{H-free}}^{\sqcup}$	$N_{\text{H-free}}^{*}$
1	$\begin{pmatrix}4\\2\end{pmatrix}_8$	2	1	2	1
2	$\begin{pmatrix}5\\2\end{pmatrix}_8$	3	1	3	1
3	$\begin{pmatrix}5\\64\end{pmatrix}_8$	4	1	5	1
4	$\begin{pmatrix}44\\32\end{pmatrix}_8$	5	2	5	1
5	$\begin{pmatrix}62\\35\end{pmatrix}_8$	6	2	7	3
6	$\begin{pmatrix}51\\664\end{pmatrix}_8$	7	4	8	2
7	$\begin{pmatrix}344\\532\end{pmatrix}_8$	8	6	9	2
8	$\begin{pmatrix}622\\575\end{pmatrix}_8$	8	1	10	4
9	$\begin{pmatrix}355\\6244\end{pmatrix}_8$	9	1	11	2
10	$\begin{pmatrix}3576\\6322\end{pmatrix}_8$	10	3	12	2

† : Total encoder memory

‡ : Minimum free branch distance

$^{\triangle}$: Number of codewords with branch distance $d_{\text{B-free}}$

$^{\sqcup}$: Free Hamming distance

* : Number of codewords with Hamming distance $d_{\text{H-free}}$

Note: The code generators have been listed in octal, where the octal representation of xyz is $4 \cdot x + 2 \cdot y + z$ and x, y and z denote 3 binary bits.

Table 2 Optimum Branch Distance Rate 2/3 Codes

γ[†]	G	$d_{\text{B-free}}$[‡]	$N_{\text{B-free}}$[△]	$d_{\text{H-free}}$[⊔]	$N_{\text{H-free}}$[*]
2	$\begin{pmatrix} 6 & 2 & 6 \\ 2 & 4 & 4 \end{pmatrix}_8$	2	4	3	2
4	$\begin{pmatrix} 0 & 4 & 3 \\ 7 & 5 & 0 \end{pmatrix}_8$	3	5	3	1
6	$\begin{pmatrix} 0 & 54 & 64 \\ 54 & 74 & 14 \end{pmatrix}_8$	4	7	6	3
8	$\begin{pmatrix} 76 & 26 & 46 \\ 64 & 0 & 36 \end{pmatrix}_8$	5	14	6	1
10	$\begin{pmatrix} 75 & 57 & 0 \\ 66 & 64 & 55 \end{pmatrix}_8$	6	30	6	1

[†] : Total encoder memory

[‡] : Minimum free branch distance

[△] : Number of codewords with branch distance $d_{\text{B-free}}$

[⊔] : Free Hamming distance

[*] : Number of codewords with Hamming distance $d_{\text{H-free}}$

Note: The code generators have been listed in octal, where the octal representation of xyz is $4 \cdot x + 2 \cdot y + z$ and x, y and z denote 3 binary bits.

Table 3 Optimum Branch Distance Rate 3/4 Codes

$\gamma^{\dagger}$	G	$d_{\text{B-free}}{}^{\ddagger}$	$N_{\text{B-free}}{}^{\triangle}$	$d_{\text{H-free}}{}^{\sqcup}$	$N_{\text{H-free}}{}^{*}$
3	$\begin{pmatrix} 0 & 6 & 6 & 2 \\ 6 & 6 & 2 & 4 \\ 6 & 2 & 2 & 2 \end{pmatrix}_8$	2	11	3	3
6	$\begin{pmatrix} 7 & 1 & 0 & 4 \\ 5 & 7 & 1 & 7 \\ 0 & 5 & 6 & 7 \end{pmatrix}_8$	3	16	5	8
9	$\begin{pmatrix} 74 & 2 & 34 & 0 \\ 44 & 7 & 74 & 74 \\ 54 & 0 & 4 & 74 \end{pmatrix}_8$	4	30	5	1

† : Total encoder memory

‡ : Minimum free branch distance

$^{\triangle}$: Number of codewords with branch distance $d_{\text{B-free}}$

$^{\sqcup}$: Free Hamming distance

* : Number of codewords with Hamming distance $d_{\text{H-free}}$

Note: The code generators have been listed in octal, where the octal representation of xyz is $4 \cdot x + 2 \cdot y + z$ and x, y and z denote 3 binary bits.

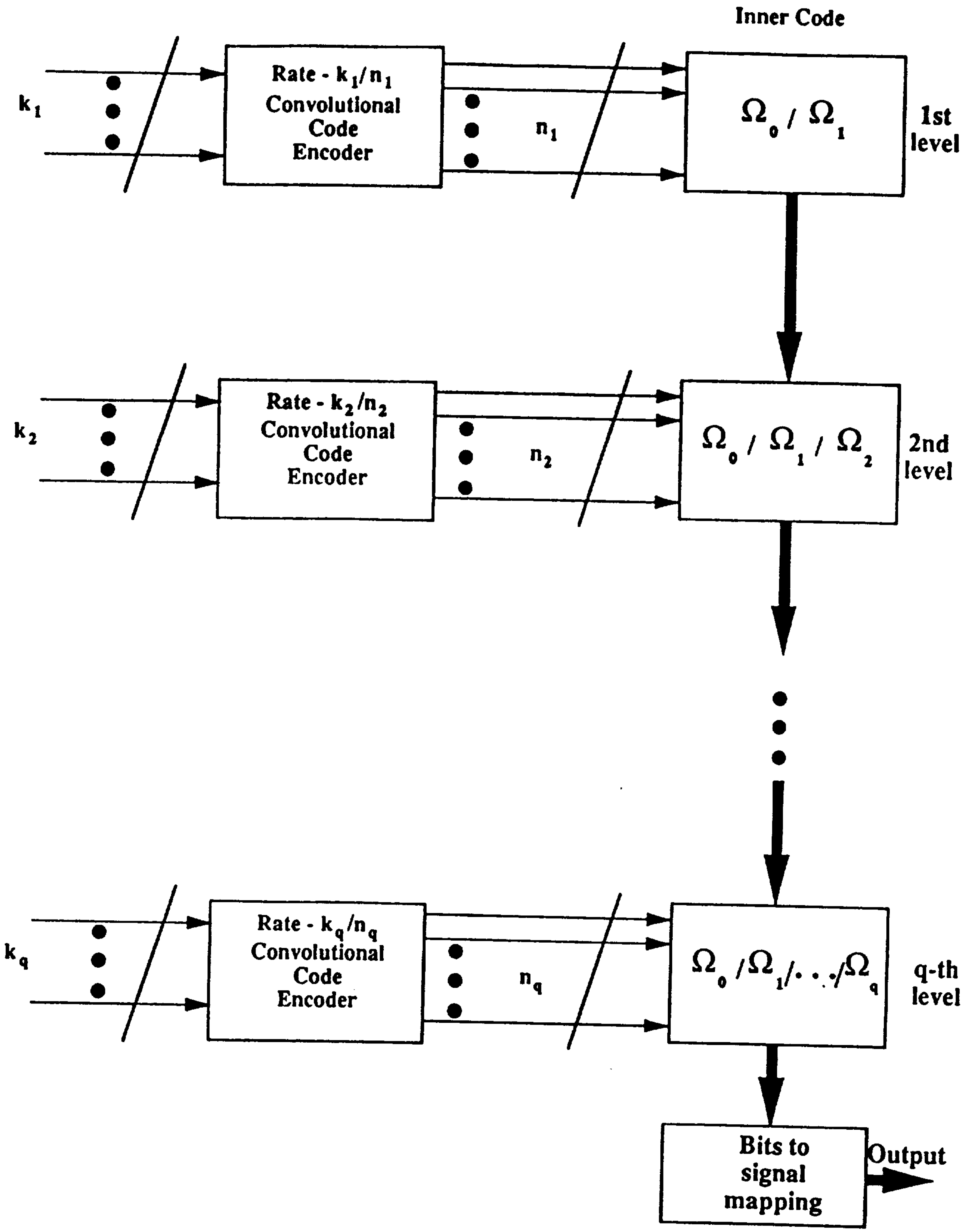
Inner Code
k_1
Rate - k_1/n_1 Convolutional Code Encoder
n_1
Ω_0 / Ω_1
1st level
k_2
Rate - k_2/n_2 Convolutional Code Encoder
n_2
$\Omega_0 / \Omega_1 / \Omega_2$
2nd level
k_q
Rate - k_q/n_q Convolutional Code Encoder
n_q
$\Omega_0 / \Omega_1 / \ldots / \Omega_q$
q-th level
Bits to signal mapping
Output

Figure 1   A multi-level concatenated TCM system

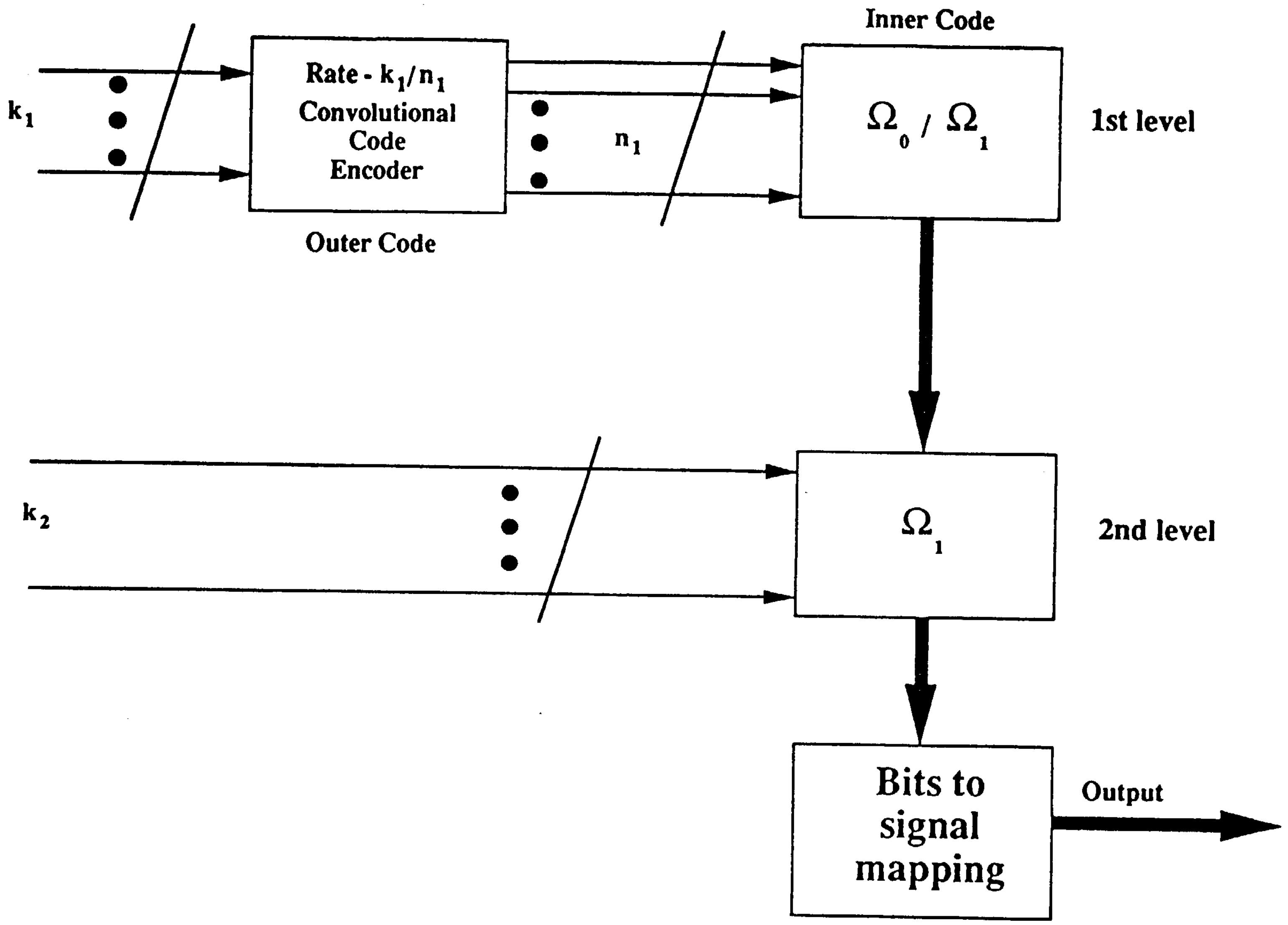

Figure 2 A two-level concatenated TCM system

Part II

CODES FOR THE AWGN AND FADING CHANNELS

Abstract

In this paper, we will use the construction technique proposed in the previous part to construct multi-dimensional TCM codes for both the AWGN and the fading channels. Analytical performance bounds and simulation results show that these codes perform very well and achieve significant coding gains over uncoded reference modulation systems. In addition, the proposed technique can be used to construct codes which have a performance/decoding complexity advantage over the codes listed in literature.

1. Introduction

As was pointed out in part one of the paper, for modulation codes over the AWGN channel, the main parameter of interest is the minimum squared Euclidean distance between the transmitted code sequences and the number of nearest neighbors. Details on the above parameters are available in [1,2] and as such, we will not reiterate these design considerations here. The aforementioned design considerations will be the basis of construction of the modulation codes for the AWGN channel in this paper.

If the channel is changed to a fading channel, most codes designed for the AWGN channel no longer perform well, simply because the design parameters of a modulation code which need to be optimized for the fading channel are different from that for the AWGN channel. For the fading channel, we shall consider two scenarios. For the first case, we shall consider

the Rayleigh fading channel with slow fading, coherent detection, no channel state information, independent symbol fading and minimum squared Euclidean distance as the decoding metric. These assumptions have been considered, so as to enable us to compare our codes with the ones listed in literature. Examples 3 and 4 construct codes for this scenario. For the second case, we consider the MSAT channel with light shadowing. Example 5 constructs a code for this case.

We would like to add, that the code construction technique is universal and is by no means restricted by the aforementioned assumptions. For the fading channels in general, the error performance of a code primarily depends on its minimum symbol distance, minimum product distance and path multiplicity. It depends on the minimum squared Euclidean distance to a lesser extent. Detailed discussion on these parameters of interest is given in [3, 4] and as such, we will not reiterate these design considerations here. The dominant parameter of interest is however the minimum symbol distance, and as such we will concentrate on optimizing this parameter, when we construct codes for the fading channel.

This paper is organized as follows: In section 2 of this paper, we will derive general analytical bounds on the performance of the modulation codes using the multi-stage decoding techniques proposed in part one of this paper. In section 3, we will construct examples using the proposed technique and compare them with the codes listed in literature.

2. Performance Analysis

In this section, we will derive a general expression for the bit-error-probability of the multi-dimensional TCM codes decoded using the multi-stage technique proposed in section 5 of part 1.

For $1 \leq i \leq q$, let X_i be a random variable, where the value of X_i denotes the number of bit errors at the i-th decoding stage at a particular time instant t. Hence, $0 \leq X_i \leq k_i$. Then, the bit-error-probability of the multi-dimensional TCM code, denoted $P_b(e)$, is:

$$P_b(e) = \mathbf{E}\ (\sum_{i=1}^{q} X_i)/\sum_{i=1}^{q} k_i = (\mathbf{E}\ (X_1)\ +\ \mathbf{E}\ (X_2)\ +\ \cdots\ + \mathbf{E}\ (X_q))/\sum_{i=1}^{q} k_i \qquad (2.1)$$

where $\mathbf{E}\ (\cdot)$ denotes the expectation operator. For $2 \leq i \leq q$, $\mathbf{E}\ (X_i)$ can be broken up into two terms, the first one being the expected number of errors at the i-th stage assuming that the previous $i-1$ stages of decoding are correct and the second one being the expected number of errors at the i-th stage due to erroneous decoding at either one of the previous $i-1$ stages of decoding, i.e., the error propagation term. Hence,

$$\mathbf{E}\ (X_i) \leq (\mathbf{E}\ (X_i)|_{\text{error propagation}} \cdot p_{E_i}) + \mathbf{E}\ (X_i)|_{\text{i-th stage error}} \tag{2.2}$$

where $\mathbf{E}\ (X_i)|_{\text{error propagation}}$ denotes the error propagation term, p_{E_i} denotes the probability of error propagation from the previous stages and $\mathbf{E}\ (X_i)|_{\text{i-th stage error}}$ denotes the term due to erroneous decoding at the i-stage, assuming that the previous $i-1$ stages of decoding are correct. Hence, (2.1) can be rewritten in the following form:

$$P_b(e) \leq (\sum_{i=2}^{q}(\mathbf{E}\ (X_i)|_{\text{error propagation}} \cdot p_{E_i}) \ + \ \sum_{i=1}^{q}(\mathbf{E}\ (X_i)|_{\text{i-th stage error}}))/\sum_{i=1}^{q} k_i \tag{2.3}$$

Except for a few specific cases, it is not possible to obtain a general expression for the expected number of bits in error due to error propagation. The expected number of bits in error due to error propagation, depend on both the choice of the inner codes as well as the outer codes, as will be shown in the examples to be discussed later in this paper. As such, we will therefore derive a general expression for the rest of the terms in (2.3).

Let $\mathbf{V}$ be the transmitted code sequence. Using (3.12, part 1) $\mathbf{V}$ can be written in the form, $\lambda(\phi_1(\mathrm{v_1}) + \phi_2(\mathrm{v_2}) + \cdots + \phi_q(\mathrm{v_q}))$, where $\mathrm{v_i}$ for $1 \leq i \leq q$ denotes a code sequence in the convolutional code at the i-th stage, C_i.

For $1 \leq i \leq (q-1)$, let us consider the term $\mathbf{E}\ (X_i)|_{\text{i-th stage error}}$. Recall from section 5 (part 1), that at the i-th stage of decoding, we form the trellis $\lambda(\tilde{C}_i)$, where a code sequence in $\lambda(\tilde{C}_i)$ is of the form, $\lambda(\phi_1(\hat{\mathrm{v}}_1) + \phi_2(\hat{\mathrm{v}}_2) + \cdots + \phi_{i-1}(\hat{\mathrm{v}}_{i-1}) + \phi_i(\mathrm{u_i}) + \omega_i)$, where $\mathrm{u_i}$ is a code sequence in the convolutional code at the i-th level, C_i; ω_i is sequence of points from Ω_i and for $1 \leq j \leq (i-1)$, $\hat{\mathrm{v}}_j$ denotes the estimate of $\mathrm{v_j}$. Since we are considering the term $\mathbf{E}\ (X_i)|_{\text{i-th stage error}}$, $\hat{\mathrm{v}}_j = \mathrm{v_j}$ for $1 \leq j \leq (i-1)$. Also, since C_i is a linear code, the code sequence $\mathrm{u_i}$ can be written in the form, $\mathrm{u_i} = \mathrm{v_i} + \mathrm{e}$, where e is code sequence in C_i. As such, any code sequence in $\lambda(\tilde{C}_i)$ can be rewritten in the form, $\hat{\mathrm{V}} = \lambda(\phi_1(\mathrm{v}_1) + \phi_2(\mathrm{v}_2) + \cdots + \phi_{i-1}(\mathrm{v}_{\mathrm{i}-1}) + \phi_i(\mathrm{v}_\mathrm{i} + \mathrm{e}) + \omega_i)$. Say, that the decoder at the i-th stage of decoding decodes the code sequence associated with the convolutional code to be $\mathrm{v}_\mathrm{i} + \mathrm{e}$, and let the probability that the event occurs be p_e. The exact expressions for p_e can be found in [1, 2] for the AWGN channel and in [3] for the Rayleigh fading channel. Let I_e denote the number of non-zero information bits associated with the sequence **e**. Then the expected number of bits in error (per decoding time instant) due to the sequence **e** is $I_\mathrm{e} \cdot p_\mathrm{e}$. Since **e** is any arbitrary code sequence in the convolutional code C_i, the total number of bits in error at the i-th stage, $\mathrm{E}\ (X_i)|_{\text{i-th stage error}}$ is obtained by considering all the possible code sequences and adding up all the $I_\mathrm{e} \cdot p_\mathrm{e}$ terms, i.e.,

$$\mathrm{E}\ (X_i)|_{\text{i-th stage error}} \leq \sum_{\mathrm{e} \in \mathrm{C}_i} I_\mathrm{e} \cdot p_\mathrm{e} \tag{2.4}$$

where C_i denotes the set of all the code sequences in the convolutional code, C_i.

Special Case - AWGN Channel: For the results derived above, let us consider the special case when the channel is AWGN. Let **V** be the transmitted code sequence and let $\hat{\mathbf{V}}$ be the decoded code sequence. Both these sequences have the form as given earlier. Let D_e^2 denote the minimum squared Euclidean distance between **V** and $\hat{\mathbf{V}}$. Since v_j for $1 \leq j \leq (i-1)$ is arbitrary, D_e^2 has been taken to be the minimum over all possible transmitted code sequences for a fixed e. This is the worst case scenario, and as such the minimum squared Euclidean distance D_e^2 gives us an upper bound on the performance of the code. Also, let N_e be the number of codewords at a squared Euclidean distance of D_e^2 from **V**. The probability that **V** is decoded incorrectly depends upon both D_e^2 as well as N_e [2]. The code sequences v_i and **e** can be written in the general form, $\mathrm{v}_\mathrm{i} = (v_{i,1}, v_{i,2}, \cdots, v_{i,p}, \cdots)$ and $\mathrm{e} = (e_1, e_2, \cdots, e_p, \cdots)$ where $v_{i,p}$ and e_p for $1 \leq p \leq \infty$ denotes the output sequence (n_i bits) of v_i and **e** respectively at the p-th time instant. The minimum squared Euclidean distance between $\hat{\mathbf{V}}$ and **V** at the p-th time instant depends only on e_p and let this squared Euclidean distance be denoted by $D_{e_p}^2$. Also, let N_{e_p} be the corresponding number of nearest neighbors [2]. Then, $D_\mathrm{e}^2 = \sum_{p=1}^{\infty} D_{e_p}^2$ and $N_\mathrm{e} = \prod_{p=1}^{\infty} N_{e_p}$. D_e^2 and N_e can be evaluated using the technique proposed in [2].

$\mathrm{E}\ (X_q)|_{\text{q-th stage error}}$ depends on whether the q-th level of encoding uses a convolutional code or is left uncoded. If a convolutional code is used at the q-th level, then the expressions for $\mathrm{E}\ (X_q)|_{\text{q-th stage error}}$ are the same as those derived above. However, if the q-th level is left uncoded then $\mathrm{E}\ (X_q)|_{\text{q-th stage error}}$ can be upper bounded as: $\mathrm{E}\ (X_q)|_{\text{q-th stage error}} \leq \mathrm{BER}_q \cdot k_q$, where BER_q denotes the decoding error probability (i.e., the block error probability) for the last stage of decoding, i.e., the block of k_q bits at the q-th stage of decoding would be declared to be in error if at least one of the bits is in error. The block error probability would depend on the decoding algorithm used at the q-th stage, i.e., single-stage or multi-stage. The block error probability can be calculated using results of [5].

A very interesting and special case of the results derived above occurs when $q = 2$ and the second level outer code is left uncoded, as shown in figure 2 (part 1). For this special case, we can get a closed form expression for $P_b(e)$. Using (2.1) and (2.2), $P_b(e)$ can be written in the form,

$$P_b(e) \leq ((\sum_{i=1}^{2} \mathrm{E}\ (X_i)|_{\text{i stage error}}) + (\mathrm{E}\ (X_2)|_{\text{error propagation}}) \cdot p_{E_2})/(k_1 + k_2), \qquad (2.5)$$

$\mathrm{E}\ (X_1)|_{\text{1-st stage error}}$ can be derived using (2.4). $\mathrm{E}\ (X_2)|_{\text{2-nd stage error}}$ can be upper bounded as, $\mathrm{E}\ (X_2)|_{\text{2-nd stage error}} \leq \mathrm{BER}_2 \cdot k_2$. Let $\mathbf{V}$ be the transmitted code sequence. Then, using (3.12, part 1), $\mathbf{V}$ can be written in the form, $\lambda(\phi_1(\mathbf{v_1}) + \omega_2)$ where $\mathbf{v_1}$ is a code sequence in the convolutional code used at the first level, C_1 and ω_2 is a sequence of points from Ω_2. Let the decoded code sequence associated with the convolutional code be $\mathbf{v_1} + \mathbf{e}$, where $\mathbf{e}$ is a code sequence in C_1. $p_{\mathbf{e}}$ gives us the corresponding probability of this event. Let $w_b(\mathbf{e})$ denote the branch weight of $\mathbf{e}$. Hence, the error sequence $\mathbf{e}$ will cause at most $w_b(\mathbf{e})$ blocks of k_2 bits at the second stage to be in error, i.e., the number of bits in error at the second stage of decoding, due to the error sequence $\mathbf{e}$ is $\leq k_2 \cdot w_b(\mathbf{e})$. Using arguments similar to those used to derive (2.4), $(\mathrm{E}\ (X_2)|_{\text{error propagation}} \cdot p_{E_2})$ can be upper bounded as, $(\mathrm{E}\ (X_2)|_{\text{error propagation}} \cdot p_{E_2}) \leq \sum_{\mathbf{e} \in \mathrm{C}_1} k_2 \cdot w_b(\mathbf{e}) \cdot p_{\mathbf{e}}$.

3. Examples

Examples 1 and 2 construct codes for the AWGN channel, examples 3 and 4 construct codes for the Rayleigh fading channel and example 5 constructs a code for the light shadowed mobile satellite communication (MSAT) channel. In the following, we will use (n, k, d) to denote a linear block code of length n, dimension k and minimum distance d.

Example 1: Consider the case of $m = 8, q = 2$ and choose $S =$ 8PSK. Hence $\ell = 3$. Figure 1 shows the two-dimensional 8PSK signal constellation of unit energy, in which each signal point is uniquely labeled with 3 bits, abc, where a is the first labeling bit and c is the last labeling bit. The labeling is done through signal partitioning process [1]. Choose $C_{0,1} = (8,4,4)$ Reed-Muller (RM) code, $C_{0,2} = C_{1,2} = (8,7,2)$ code, $C_{0,3} = C_{1,3} = (8,8,1)$ code and $C_{1,1} = (8,1,8)$ code. The minimum squared Euclidean distance of $\Lambda_0 = \lambda(\Omega_0)$ is 2.344 and for $\Lambda_1 = \lambda(\Omega_1)$ is 4.0 [5]. The encoder structure will be the same as that in figure 2 (part 1). A rate-2/3 code will be used at the first level. Two choices will be considered for the convolutional code at the first level. The first choice is the 4-state, $d_{\text{B-free}} = 2$ code from Table 2 (part 1) and the second choice is the 16-state, $d_{\text{B-free}} = 3$ code from Table 2 (part 1). The phase invariance of the resulting code is the same for both the choices and is 45° and can be derived by a straightforward application of theorem 7 (part 1). The spectral efficiency is also the same for both the choices and is equal to $(16+2)/8 = 2.25$ bits/symbol. The mapping ϕ_1 used is linear. Details of ϕ_1 have been omitted due to lack of space. The following gives a detailed discussion for both the choices :

4 state: The minimum squared Euclidean distance of the code is (refer theorem 5, part 1): $\min\{4.0, 2.344 \cdot 2\} = 4.0$. Using (3.12, part 1), any code sequence in the super trellis can be written in the form, $\lambda(\phi_1(\mathbf{v}_1) + \omega_1)$ where $\mathbf{v}_1$ is code sequence in the 2/3- rate convolutional code used at the first level, and ω_1 is sequence of points from Ω_1. As such, the super trellis for this code is isomorphic to the trellis of the convolutional encoder used at the first level, with each branch of the trellis consisting of 2^{16} parallel transitions corresponding to the 2^{16} elements of Ω_1. Ω_1 has a 4-state, 8-section trellis diagram [5]. Each branch of the super trellis can be expressed in the form, $\lambda(\omega_0 + \Omega_1)$ where $\omega_0 \in [\Omega_0/\Omega_1]$. Hence, each branch of the super trellis has a 4-state, 8-section trellis, which is **isomorphic** to the trellis of Ω_1. Standard Viterbi decoding can be used on every branch of super trellis using this 4-state, 8-section **isomorphic** trellis to find the most probable parallel transition. The trellis of the overall multi-dimensional code can thus be viewed as a **nested trellis diagram**, i.e., a trellis within a trellis.

A reduction in the decoding complexity can be achieved by using the **multi-stage decoding** algorithm proposed in section 5 (part 1). The decoding now proceeds in two stages. Let $\mathbf{V}$ be the transmitted code sequence. Using (3.12, part 1) $\mathbf{V}$ can be written in the form, $\lambda(\phi_1(\mathrm{v}_1) + \omega_1^{\mathrm{tr}})$ where v_1 is a code sequence in the convolutional code C_1 used at the first level, and ω_1^{tr} is a sequence of points from Ω_1. At the first stage of decoding, we form the trellis $\tilde{C}_1^{\mathrm{sup}}$ where any code sequence in $\tilde{C}_1^{\mathrm{sup}}$ can be written in the form $\phi_1(\mathrm{u}_1) + \omega_1^{\mathrm{sup}}$, where ω_1^{sup} denotes a sequence of points from Ω_1^{sup} and u_1 is a code sequence in C_1. The details of how the trellis $\tilde{C}_1^{\mathrm{sup}}$ is formed were mentioned in section 5 (part 1). Ω_1^{sup} is chosen to be: $\Omega_1^{\mathrm{sup}} = (8,1,8)*(8,8,1)*(8,8,1)$ which has a very simple 2-state trellis structure. On the other hand, Ω_1 has a 4-state 8-section trellis diagram which is more complex than the trellis structure of Ω_1^{sup}. This helps in reducing the closest coset decoding complexity associated with the first stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\lambda(\tilde{C}_1^{\mathrm{sup}})$ to obtain an estimate of v_1, denoted $\hat{\mathrm{v}}_1$. This completes the first stage of decoding.

At the second stage of decoding, we construct the trellis $\tilde{C}_2$ where a code sequence in $\tilde{C}_2$ is of the form, $\phi_1(\hat{\mathrm{v}}_1)+\omega_1$ where ω_1 denotes a sequence of points from Ω_1. Consider the p-th time instant. The structure of $\tilde{C}_2$ at the p-th time instant is of the form, $\tilde{C}_{2,p} = \phi_1(\hat{\mathrm{v}}_{1,p})+\Omega_1$ where $\hat{\mathrm{v}}_{1,p}$ is the component of $\hat{\mathrm{v}}_1$ at the p-th time instant. This trellis $\tilde{C}_{2,p}$ is **isomorphic** to the trellis Ω_1 and this trellis can be used to obtain an estimate of $\omega_{1,p}^{\mathrm{tr}}$, where $\omega_{1,p}^{\mathrm{tr}}$ is the term in ω_1^{tr} corresponding to the p-th time instant.

The decoding complexity associated with the second stage of decoding can be further reduced by using the 3-stage decoding technique for Ω_1 proposed by Sayegh [6] and Tanner [7]. We will carry out the second stage of decoding using the 3-stage decoding technique mentioned above [2].

The multi-stage decoding algorithm does lead to a slight degradation in performance, however, as will be shown in the performance curves, the loss is negligible as compared to the reduction in complexity. The following gives the number of computations associated with both the optimal and the multi-stage decoding algorithm for the 4-state trellis. The complexity calculation for the multi-stage decoding algorithm has been carried out assuming the 3-stage decoding for the second stage, as mentioned above.

Computation Complexity - Optimal Decoding Algorithm: $\gamma_1 = 2$ and $k_1 = 2$.

The branch decoding complexity B_{C_1} is: (1) since there are eight 8PSK points per branch, the distance computation complexity per branch is 64; (2) survivor calculation for the parallel branch transitions in Ω_1 requires 32 compares; and (3) the Viterbi decoding for Ω_1 requires 52 adds and 27 comparison to calculate the final survivor (assuming the survivor for the parallel transitions has been found). Since there are 8 cosets, the total complexity is 416 adds and 216 compares, i.e., B_{C_1} = 416 adds + 248 compares + 64 distance computations. Hence, total complexity is 54 adds + 32.5 compares + 8 distance computations per 2-dimensions.

Computation Complexity - Multi-stage Decoding Algorithm: $\gamma_1 = 2$ and $k_1 = 2$.

The branch decoding complexity is:

First stage of decoding: (1) there are eight 8PSK points per branch, hence the distance computation complexity per branch is 64; (2) the sub-optimal distance estimates [7] require 48 compares; (3) Viterbi decoding of Ω_1^{sup} requires 14 adds and 1 compare. Since there are 8 cosets, the total complexity is 112 adds and 8 compares.

Second stage of decoding: (1) the multi-stage decoding technique requires 26 adds and 13 compares. Hence, total complexity is 19.25 adds + 10.125 compares + 8 distance com-

[2] Note, the first stage of the 3-stage decoding process for Ω_1 can actually be combined with the first stage of decoding of the TCM code, i.e. the stage which uses the trellis $\bar{C}_1^{\text{sup}}$.

putations per 2-dimensions.

Figure 2 shows the simulation results of the bit-error-performance of both the optimal and the multi-stage decoding algorithm. An upper bound on the bit-error-rate of the proposed code is also shown in figure 2. Details of the bound have been omitted due to lack of space. Also shown in the figure is the bit-error-performance of a hypothetical uncoded PSK system of the same spectral efficiency [9].

Figure 2 shows that the multi-stage and optimal decoding curves converge around $E_b/N_o =$ 8dB, and the performance of the optimal curve is only slightly better at low SNR. The proposed code achieves a coding gain of 2.8 dB at the decoded bit-error-rate of 10^{-6} over the uncoded reference system of the same spectral efficiency [9]. In addition, the decoding complexity of the optimal decoding algorithm is roughly about 3 times the decoding complexity of the sub-optimal one.

Pietrobon et. al. do not have a comparable code over 8×2-dimensions, hence comparison will be made with a 4×2-dimensional code over 8PSK with $\gamma = 2$ and phase invariance $= 45°$. Spectral efficiency of this code is 2.25 bits/symbol, same as that of the proposed code. The performance curve of this code, taken from [11], has also been shown in the figure. The complexity of the Pietrobon code is 24 adds + 17 compares + 8 distance distance computations per 2-dimensions. As can be seen from the figure, the proposed code outperforms the Pietrobon code by roughly 0.4 dB at $4 \cdot 10^{-6}$ bit-error-rate, and in addition, the complexity of the proposed code with multi-stage decoding is less than that of the Pietrobon code.

16 states: The minimum squared Euclidean distance of the code is (refer theorem 5, part 1) $\min\{4.0, 2.344 \cdot 3\} = 4.0$. The super-trellis in this case is very similar to the 4-state trellis discussed above, with the only difference that the 4-state convolutional code at the first level, has been replaced by the 16-state trellis. Both the optimal and the multi-stage decoding techniques will be investigated for this case also. The complexity associated with the optimal and the multi-stage decoding technique are:

Computation Complexity - Optimal Decoding Algorithm: $\gamma_1 = 4$ and $k_1 = 2$.

The branch decoding complexity B_{C_1} is the same as the 4-state case. Therefore, total complexity is 60 adds + 37 compares + 8 distance computations per 2-dimensions.

Computation - Complexity - Multi-stage Decoding Algorithm: $\gamma_1 = 4$ and $k_1 = 2$.

The branch decoding complexity is the same as the 4-state case. Therefore, total complexity is 25.25 adds + 14.625 compares + 8 distance computations per 2-dimensions.

Figure 3 shows the bit-error-performance of the both the optimal and the sub-optimal-decoding algorithm. An upper bound on the bit-error-rate of the proposed code using the multi-stage decoding algorithm is also shown in figure 3.

Figure 3 shows that the multi-stage and the optimal decoding curves exhibit the same characteristics as the 4-state case. The two curves converge around $E_b/N_0 = 6.54$ dB, and the performance of the optimal curve is only slightly better than the optimal curve at low SNR. The proposed code achieves a coding gain of 3.2 dB at the decoded bit-error-rate of 10^{-6} over the uncoded reference system of the same spectral efficiency [9]. In addition, the decoding complexity of the optimal decoding algorithm is roughly about 2.5 times the decoding complexity of the multi-stage one.

Pietrobon et. al. do not have a comparable code over 8×2-dimensions, hence comparison will be made with a 4×2-dimensional code over 8PSK with $\gamma = 3$ and phase invariance = 45°. Spectral efficiency of this code is 2.25 bits/symbol, same as that of the proposed code. The performance curve of this code, taken from [8], has also been shown in the figure. The complexity of this code is 48 adds + 32 compares + 8 distance distance computations per 2-dimensions. The performance of the proposed code is slightly better than the Pietrobon code and in addition the complexity of the Pietrobon code is about 2 times higher than that of the proposed code with multi-stage decoding.

The 16-state proposed code with the multi-stage decoding algorithm achieves better performance than the 4-state proposed code with the multi-stage decoding algorithm at the cost of slightly increased decoding complexity. The improvement in performance is due to the higher minimum squared Euclidean distance of the first decoding stage of the 16-state code. This leads to better performance at the first decoding stage and as a result reduced error propagation onto the second decoding stage.

Example 2: Consider the case of $m = 16, q = 3$ and choose $S =$ 8PSK. Hence $\ell = 3$. Choose $C_{0,1} = (16,4,8)$ code. This code is obtained from the first order Reed-Muller code of length 16, by removing the all ones vector from the generator matrix of the $(16,5)$ code. Choose $C_{2,2} = (16,11,4)$ RM code, $C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = C_{2,3} = (16,15,2)$ code and $C_{1,1} = C_{2,1} = (16,0,\infty)$ code, i.e., the code consisting of just the all zero codeword. The minimum squared Euclidean distance for $\Lambda_0 = \lambda(\Omega_0)$ is 4.0, for $\Lambda_1 = \lambda(\Omega_1)$ is 4.0 and for $\Lambda_2 = \lambda(\Omega_2)$ is 8.0 [5]. A rate-3/4 code with 64-states (second code in table 3 of part 1) will be used at the first level. Let us call this code C_1. The same rate-3/4 code used at the first level will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 90°. The spectral efficiency is equal to $(3+3+26)/16 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 used at the first and second encoding levels respectively have been chosen to be linear. The minimum squared Euclidean distance of the code is at least (refer theorem 5, part 1), $\min\{8.0 \ , \ 3 \cdot 4.0 \ , \ 3 \cdot 4.0\} = 8.0$. Note, that the theorem gives the minimum squared Euclidean distance associated with the first encoding stage to be at least 12.0. A quick verification of the partitions given above show that the minimum squared Euclidean distance is actually $3 \times 8 \times 0.586 = 14.064$. This is obtained by considering the squared Euclidean distance due to the (16, 4) code of Ω_0 and multiplying it by the free branch distance of C_1.

Optimal decoding of the multi-dimensional code would require a trellis with $2^6 \cdot 2^6 = 2^{12}$ states. Optimal decoding of the code using this 4096 state trellis would be extremely complex, and as such we will focus on the multi-stage decoding technique proposed in section 5 (part 1). The multi-stage decoding of the multi-dimensional code proceeds in 3 stages.

Let $\mathbf{V}$ be the transmitted code sequence. Using (3.12, part 1) $\mathbf{V}$ can be expressed in the form, $\lambda(\phi_1(\mathrm{v}_1) + \phi_2(\mathrm{v}_2) + \omega_2)$ where v_1 is a code sequence in the 64-state convolutional code C_1, $\mathbf{v_2}$ is a code sequence in the 64-state convolutional code C_2 and ω_2 is a sequence of points from Ω_2.

First stage of decoding: To simplify the trellis decoding complexity associated with the first stage of decoding, instead of forming the trellis $\tilde{C}_1$ we form the trellis $\tilde{C}_1^{\mathrm{sup}}$, where any code sequence in $\tilde{C}_1^{\mathrm{sup}}$ can be written in the form (refer section 5, part 1), $\phi_1(\mathrm{u}_1) + \omega_1^{\mathrm{sup}}$ where ω_1^{sup} is a sequence of points from Ω_1^{sup} and u_1 is a code sequence in C_1. Ω_1^{sup} is chosen to be, $\Omega_1^{\mathrm{sup}} = (16, 0, \infty) * (16, 16, 1) * (16, 16, 1)$. Ω_1^{sup} has a very simple 1-state trellis structure. On the other hand, Ω_1 has a 4-state trellis diagram which is more complex than the trellis structure of Ω_1^{sup}. This helps in reducing the closest coset decoding complexity associated with the first stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\tilde{C}_1^{\mathrm{sup}}$ to obtain an estimate of v_1, denoted $\hat{\mathrm{v}}_1$. This completes the first stage of decoding.

Second stage of decoding: To simplify the trellis decoding complexity associated with the second stage of decoding, instead of forming the trellis $\tilde{C}_2$, we form $\tilde{C}_2^{\mathrm{sup}}$, where any code sequence in $\tilde{C}_2^{\mathrm{sup}}$ can be written in the form (refer section 5, part 1), $\phi_1(\hat{\mathrm{v}}_1) + \phi_2(\mathrm{u}_2) + \omega_2^{\mathrm{sup}}$ where ω_2^{sup} is a sequence of points from Ω_2^{sup} and u_2 is a code sequence in C_2. Ω_2^{sup} is chosen to be, $\Omega_2^{\mathrm{sup}} = (16, 0, \infty) * (16, 11, 4) * (16, 16, 1)$. Ω_2^{sup} has a 8-state trellis structure [10]. On the other hand, Ω_2 has a 16-state trellis diagram which is more complex than the trellis structure of Ω_2^{sup}. This helps in reducing the closest coset decoding complexity associated with the second stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\tilde{C}_2^{\mathrm{sup}}$ to obtain an estimate of $\mathbf{v_2}$, denoted $\hat{\mathbf{v}}_2$. This completes the second stage of decoding.

Third stage of decoding: The third stage of decoding is identical to the second stage of decoding discussed in example 1. The three stage decoding technique proposed by Sayegh [6] and Tanner [7] is used to split up the decoding of Ω_2 into three stages. The first stage decoding of Ω_2 is trivial. Note, the second stage of the 3-stage decoding process for Ω_2 can be combined with the second stage of decoding of the multi-dimensional TCM code.

Computation Complexity - Multi-stage Decoding Algorithm: $\gamma_1 = 6, k_1 = 3, \gamma_2 = 6, k_2 = 3$. The branch decoding complexity is:

First stage of decoding: (1) the distance computation complexity per branch is 128; (2) the sub-optimal distance estimates require 96 compares; (3) Viterbi decoding of Ω_1^{sup} requires 3 adds. Since there are 16 cosets, the total complexity is 48 adds.

Second stage of decoding: (1) closest coset decoding for Ω_2^{sup} requires 184 adds + 87 compares, which is the trellis decoding complexity of the (16, 11, 4) code [10]. Since there are 16 cosets, the total complexity is 2944 adds and 1392 compares.

Third stage of decoding: (1) the multi-stage decoding technique for Ω_2 requires 58 adds and 29 compares. Note, only the decoding complexity of the (16, 15, 2) code has been taken into account. The decoding complexity of the (16, 11, 4) code is included in the second stage of decoding for reasons mentioned above. Hence, total complexity is 254.62 adds + 150.81 compares + 8 distance computations per 2-dimensions.

Figure 4 shows the simulation results of the bit-error-performance of multi-dimensional TCM code. As can be seen from the figure, the code achieves a 4.2 dB coding gain over uncoded QPSK at 10^{-6} bit error rate. An upper bound on the bit-error-rate of the proposed code using the multi-stage decoding algorithm is also shown in figure 4.

Pietrobon et. al. do not have a comparable code over 16×2-dimensions, hence comparison will be made with a 2×2-dimensional code over 8PSK with $\gamma = 7$ and phase invariance = 90°. The spectral efficiency and phase invariance of both codes is the same. This Pietrobon et. al. code is the best in performance among all the codes listed in [2] for rate 2 bits/symbol. The performance curve of this code, taken from [8], has also been shown in the figure. The complexity of the Pietrobon code is about 2 times higher than that of the proposed code, however, the proposed code has performance comparable to the Pietrobon code at high SNR.

Example 3: Consider the case of $m = 2, q = 3$ and choose $S =$ 8PSK. Hence $\ell = 3$. Choose

$C_{0,1} = C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = (2,2,1)$ code, $C_{2,3} = (2,1,2)$ code and $C_{2,1} = C_{2,2} = C_{1,1} = (2,0,\infty)$ code. The minimum symbol distance of $\Lambda_0 = \lambda(\Omega_0)$ is 1, for $\Lambda_1 = \lambda(\Omega_1)$ is 1 and for $\Lambda_2 = \lambda(\Omega_2)$ is 2 (refer section 3, part 1). The other distance parameters associated with the three block modulation codes can be found by a straightforward application of the distance theorem in [4]. A rate-1/2 code with 16-states (fourth code in Table 1 of part 1) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in Table 2 of part 1) will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 180°. The spectral efficiency is equal to $(1+2+1)/2 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 have been chosen to be linear.

The minimum symbol distance of the code is (refer theorem 6, part 1), $\min\{2\ ,\ 3 \cdot 1, 5 \cdot 1\} = 2$. Since the minimum symbol distance of the overall modulation code is the minimum symbol distance of Λ_2, hence the minimum product distance, Δ_p^2 of the modulation code is $(4.0)^2 = 16.0$ (refer [4]).

The decoding of this code is carried out in three stages and proceeds exactly as discussed in section 5 (part 1). The second and third stage of decoding can actually be combined into one single stage of decoding. The computational complexity calculated below assumes that the second and third decoding stages have been combined.

The minimum symbol distance of the first stage is chosen to be higher than the rest of the decoding stages, so as to reduce the effect of error propagation.

Computation Complexity - Multi-stage Decoding Algorithm: $\gamma_1 = 4, k_1 = 1, \gamma_2 = 4, k_2 = 2$. The branch decoding complexity is:

First stage of decoding: (1) the distance computation complexity per branch is 16; (2) the sub-optimal distance estimates require 12 compares; (3) Viterbi decoding of Ω_1 requires 1 add. Since there are 4 cosets, the total complexity is 4 adds.

Second and third stage of decoding: (1) Viterbi decoding of Ω_2 is 2 adds + 1 compares. Since there are 8 cosets, the total complexity is 16 adds and 8 compares. Therefore, total complexity is 58 adds + 42 compares + 8 distance computations per 2-dimensions.

Figure 5 shows the simulation results of the bit-error-performance of the proposed code. The performance of this code will be compared with the 16-state rate-2/3 code over 8PSK constructed by Schlegel and Costello [12] for the Rayleigh fading channel. The spectral efficiency for both codes is the same, however the Schlegel-Costello code has no phase invariance. The performance curve of the Schlegel-Costello code is also shown in figure 5. As can be seen from the figure, the proposed code outperforms the Schlegel-Costello code by about 1.6 dB at 10^{-4} bit error rate. In addition, the complexity of the Schlegel-Costello code is 64 adds + 48 compares + 8 distance computations per 2-dimensions which is slightly higher than that of the proposed code.

Example 4: Consider the case of $m = 8, q = 4$ and choose $S =$ 8PSK. Hence $\ell = 3$. Choose $C_{0,1} = C_{2,2} = C_{3,2} = C_{3,3} = (8,4,4)$ RM code, $C_{0,2} = C_{1,2} = (8,7,2)$ code, $C_{0,3} = C_{1,3} = C_{2,3} = (8,8,1)$ code and $C_{1,1} = C_{2,1} = C_{3,1} = (8,0,\infty)$ code. A rate-3/4 code with 8-states (first code in Table 3 of part 1) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in Table 2 of part 1) will be used at the second level. Let us call this code C_2. A rate-3/4 code with 64-states (second code in Table 3 of part 1) will be used at the third level. Let us call this code C_3. The phase invariance of the resulting code is 180°. The spectral efficiency is equal to $(3+2+3+8)/8 = 2$ bits/symbol. The mappings ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are chosen to be linear.

The decoding of this code is carried out in four stages and proceeds in a manner similar to that in example 2. The first stage of decoding is similar to the first stage of decoding in example 2. Ω_1^{sup} used to simplify the decoding complexity is, $\Omega_1^{\text{sup}} = (8,0,\infty) * (8,8,1) * (8,8,1)$. Ω_1^{sup} has a very simple 1-state trellis which is less complex than the 2-state trellis of Ω_1. The second and third stage of decoding is carried out exactly as described in section 5 (part 1). The fourth stage of decoding is carried out using the multi-stage decoding technique for Ω_3 (as was explained in example 1). The multi-stage decoding of Ω_3 proceeds in two stages. The first stage of decoding decodes the code $C_{3,2}$ and the second stage decodes the $C_{3,3}$ code. The decoding of $C_{3,2}$ can be merged with the second stage of decoding of the proposed code, and the decoding of $C_{3,3}$ can be merged with the third stage decoding of the proposed code. The complexity calculations given below assume that the fourth stage of decoding of the proposed code has been merged with the previous stages.

Computation Complexity - Multi-stage Decoding Algorithm: $\gamma_1 = 3, k_1 = 3, \gamma_2 = 4, k_2 = 2, \gamma_3 = 6, k_3 = 3$. The branch decoding complexity is:

First stage of decoding: (1) the distance computation complexity per branch is 64; (2) the sub-optimal distance estimates require 48 compares; (3) Viterbi decoding of Ω_1^{sup} requires 7 adds. Since there are 16 cosets, the total complexity is 112 adds.

Second stage of decoding and the 1st stage of the fourth stage of decoding: (1) closest coset decoding complexity is 36 adds and 11 compares, which is the trellis decoding complexity of the (8, 4, 4) code [10]. Since there are 8 cosets, the total complexity is 288 adds and 88 compares.

Third stage of decoding and the 2nd stage of the fourth stage of decoding: (1) closest coset decoding complexity is 36 adds and 11 compares, which is the trellis decoding complexity of the (8, 4, 4) code [10]. Since there are 16 cosets, the total complexity is 576 adds and 176 compares. Therefore, total complexity is, 202 adds + 108 compares + 8 distance computations per 2-dimensions.

Figure 6 shows the simulation results of the bit-error-performance of the proposed code. The performance of this code will be compared with the 64-state rate-2/3 code over 8PSK constructed by Schlegel and Costello [12] for the Rayleigh fading channel. The spectral efficiency for both codes is the same, however the Schlegel-Costello code has no phase invariance. The performance curve of the Schlegel-Costello code is also shown in figure 6. As can be seen from the figure, the proposed code outperforms the Schlegel-Costello code by about 1.5 dB at $2 \cdot 10^{-4}$ bit error rate. In addition, the complexity of the Schlegel-Costello code is 256 adds + 192 compares + 8 distance computations per 2-dimensions which is higher than that of the proposed code.

Example 5: A statistical model for the shadowed mobile satellite channel has been devised by Loo [13-16] and this model has been used by other researchers [17-22] to study the error performance of coded modulation schemes over the MSAT channel. In Loo's model, there are three different kinds of shadowing - light, average and heavy. The corresponding Rician factors are 6.16, 5.46 and -19.33 dB, respectively. Therefore, in the shadowed MSAT channel, a coded modulation system suffers very severe distortion due to randomly changing phase and multipath fading. Especially, if the Doppler frequency shift is large due to the motion of vehicle, a coded modulation system faces the error floor phenomenon. We will assume that the carrier frequency is 870 MHz and the symbol rate is 2400 symbols/sec. Due to randomly changing phase, perfect phase synchronization is not feasible in the shadowed MSAT channel. Therefore, differentially detected 8PSK modulation is used. We assume that the speed of moving object is 92.88 miles/hr. The corresponding normalized fading bandwidth BT is 0.05 where B is the maximum Doppler frequency shift and T^{-1} is the symbol rate. To combat burst errors, a block interleaver is used for computer simulation. The size of interleaver is 512 8DPSK symbols, and the number of rows of the block interleaver is 64 and the number of columns is 8.

Consider the case of $m = 8, q = 3$. Hence, $\ell = 3$. Choose $C_{0,1} = C_{2,2} = (8,4,4)$ RM code, $C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = C_{2,3} = (8,7,2)$ code and $C_{1,1} = C_{2,1} = (8,0,\infty)$ code. A rate-3/4 code with 8-states (first code in Table 3 of part 1) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in Table 2 of part 1) will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 90°. The spectral efficiency is equal to $(3+2+11)/8 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 used at the first and second encoding levels are linear.

Decoding of the code proceeds exactly as in example 2, and as such, will not be repeated here. The complexity calculations are also very similar to example 2, and as such details will be omitted. The total complexity is, 69.25 adds + 31.63 compares + 8 distance computations per 2-dimensions.

Figure 7 shows the simulation results of the bit-error-performance of the proposed code. The performance of this code will be compared with the 16-state rate-2/3 code constructed by Schlegel and Costello [12] (this code is chosen, for lack of comparable complexity code available in literature for the shadowed MSAT channel). The spectral efficiency for both codes is the same. The performance curve of the Schlegel-Costello code is also shown in figure 7. As can be seen from the figure, the proposed code outperforms the Schlegel-Costello code by about 9.65 dB at 10^{-4} bit error rate. Also, the proposed code faces the error floor at around 1.4×10^{-5} bit error rate, whereas the Schlegel-Costello code faces an error floor around 4.8×10^{-5} bit error rate. In addition, the complexity of the Schlegel-Costello code is higher than that of the proposed code.

4. Conclusion

A simple and systematic technique of constructing multi-dimensional TCM codes using block modulation codes and convolutional codes optimized for branch distance is proposed. Bounds on the minimum squared Euclidean distance and minimum symbol distance of the multi-dimensional TCM codes are derived, along with conditions on phase invariance. A multi-stage decoding technique for the multi-dimensional TCM codes has also been proposed. Examples constructed show that the technique can be used to construct good codes which have a performance/decoding complexity advantage over the codes available in literature for both the AWGN and fading channels.

References

[1] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," **IEEE Trans. on Info. Theory**, Vol. IT-28, No. 1, pp. 55-67, Jan. 1982.

[2] S.S. Pietrobon et. al., "Trellis Coded Multi-Dimensional Phase Modulation," **IEEE Trans. on Info. Theory**, Vol. IT - 36, No. 1 , pp. 63 - 89, January 1990.

[3] D. Divsalar and M.K. Simon, "The Design of Trellis Coded MPSK for Fading Channels

: Performance Criteria," **IEEE Trans. on Comm.**, pp. 1004 - 1012, Sep. 1988.

[4] J. Wu and S. Lin, "Multilevel Trellis MPSK Modulation Codes for the Rayleigh Fading Channel," **IEEE Trans. on Comm.**, Vol. 41, No. 9, Sep. 1993.

[5] T. Kasami et. al., "On Multi-level Block Modulation Codes," **IEEE Trans. on Info. Theory**, Vol. IT - 37, No. 4 , pp. 965 - 975, July 1991.

[6] S.I. Sayegh, "A Class of Optimum Block Codes in Signal Space," **IEEE Trans. on Comm.**, Vol. COM-30, No. 10, pp. 1043-1045, Oct. 1986.

[7] R.M. Tanner, "Algebraic Construction of Large Euclidean Distance Combined Coding Modulation Systems," **Abstract of Papers, 1986 IEEE International Symposium on Information Theory**, Ann Harbor, October 6-9, 1986.

[8] D.J. Costello, Jr., private communication.

[9] R. H. Deng and D. J. Costello, Jr., "High Rate Concatenated Coding Systems Using Multi-dimensional Bandwidth Efficient Trellis Inner Codes," **IEEE Trans. on Comm.**, Vol. COM-37, pp. 1091-1096, Oct. 1989.

[10] G. D. Forney, Jr., "Coset Codes - Part II : Binary Lattices and Related Codes," **IEEE Trans. on Info. Theory**, Vol. IT - 34, No. 5, pp. 1152 - 1187, Sep. 1988.

[11] L. C. Perez and D. J. Costello, Jr., "On the Performance of Multi-dimensional Phase Modulated Trellis Codes," **NASA Technical Report # 89-10-02**, NASA Training Grant NGT-7010, Oct. 1989.

[12] C. Schlegel and D. J. Costello, Jr., "Bandwidth Efficient Coding for Fading Channels: Code Construction and Performance Analysis," **IEEE Journal on Selected Areas in Comm.**, Vol. 7, No. 9, Dec. 1989.

[13] C. Loo, "Measurements and Models of a Mobile-Satellite Link with Applications," **GLOBECOM'85**, Dec. 2-5, 1985.

[14] —, "A Statistical Model for a Land Mobile Satellite Link," in **Links, for the future (ICC'84), Science, Systems, and Services for Comm.**, P. Dewilde and C. A. May, Ed. New York : IEEE/Elsevier, North-Holland, 1984.

[15] —, "A Statistical Model for a Land Mobile Satellite Link," **IEEE Trans. Veh. Tech.**, vol. VT-34, pp. 122-127, Aug. 1985.

[16] C. Loo et. al., "Measurements and Modeling of Land-Mobile Satellite Signal Statistics," **presented at 1986 Veh. Tech.. Conf.**, May 20-22, 1986.

[17] P. J. McLane et. al., "PSK and DPSK Trellis Codes for Fast Fading, Shadowed Mobile Satellite Communication Channels," in **Pro. 1987 Int. Conf. Commun.**, June 7-10, 1987, pp. 21.1.1-21.1.6.

[18] —, "PSK and DPSK Trellis Codes for Fast Fading, Shadowed Mobile Satellite Communication Channels," **IEEE Trans. Commun.**, vol. 36, pp. 1242-1246, Nov. 1988.

[19] A. C. M. Lee and P. McLane, "Conventionally interleaved PSK and DPSK Trellis Codes for Shadow, Fast Fading, Mobile Satellite Communications Channel," **IEEE Trans. Veh. Technol.**, vol. 39, pp. 37-47, Feb. 1990.

[20] R. G. McKay et. al., "Analytical Performance Bounds on Average Bit Error Probability for Trellis Coded PSK Transmitted over Fading Channels," in **Proc. IEEE Int. Conf. Comm.**, 1989, pp. 9.2.1-9.2.7. Also to appear in **IEEE Trans. Comm.**

[21] S. H. Jamali and T. Le-Ngoc, "A new 4-state 8PSK TCM Scheme for Fast Fading, Shadowed Mobile Radio Channels," **IEEE Trans. Veh. Tech.**, pp. 216-222, Feb. 1991.

[22] —, "Performance Comparison of Different Decoding Strategies for a Bandwidth-Efficient Block-Coded Scheme on Mobile Radio Channels," **IEEE Trans. on Veh. Tech.**, Vol. 41, No. 4, pp. 505-515, Nov. 1993.

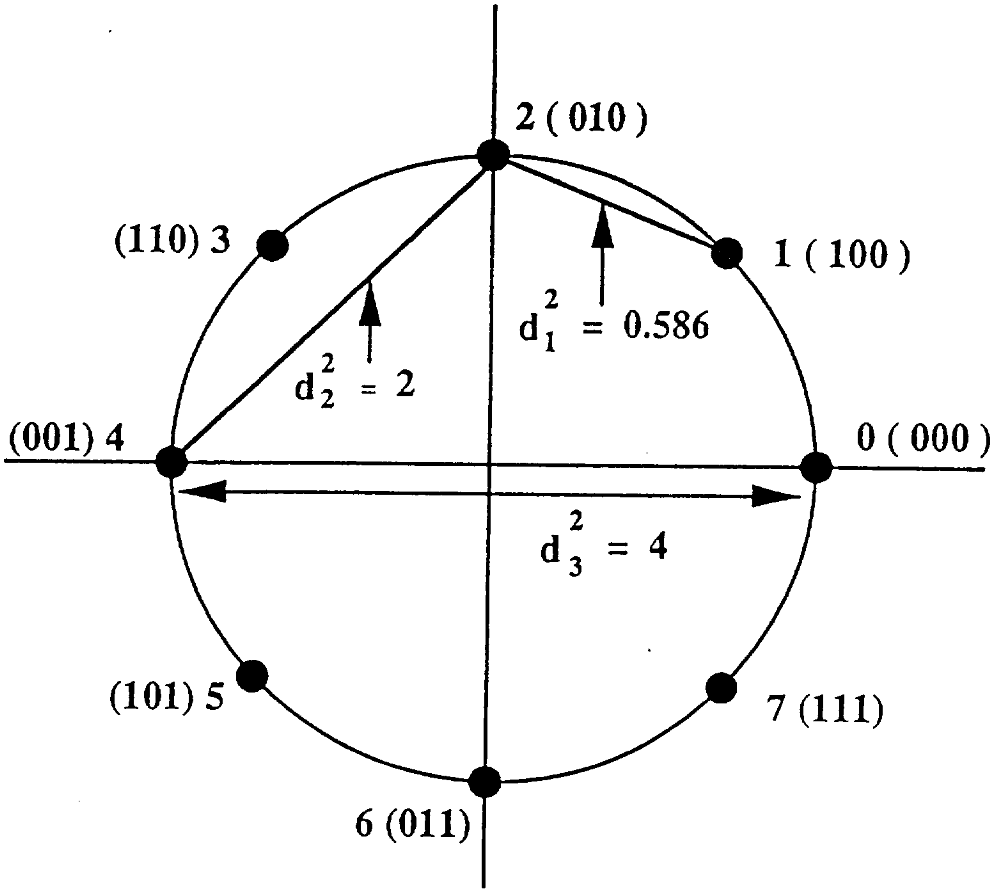

Figure 1 An 8PSK signal constellation and its signal labels

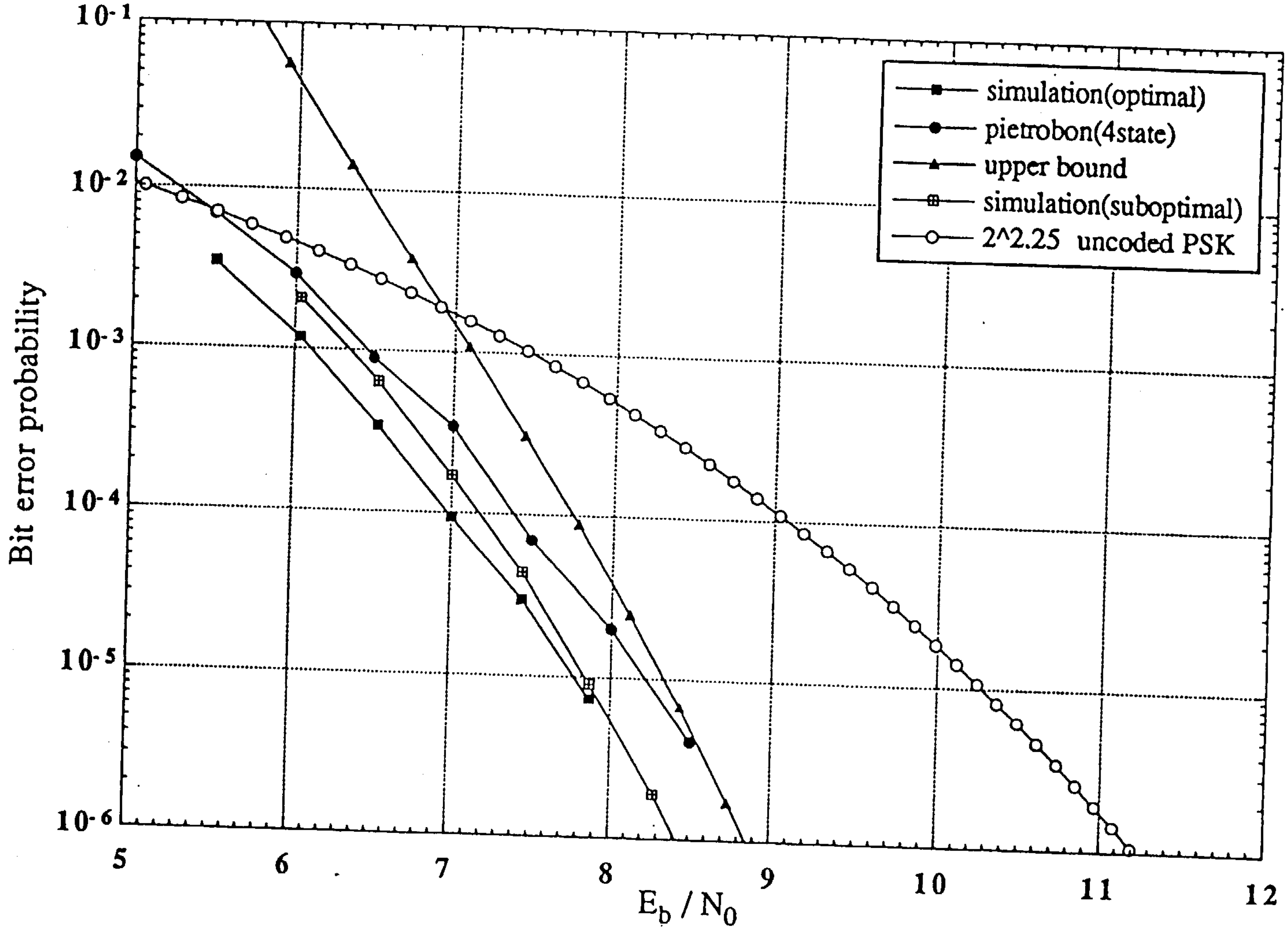

Figure 2 Bit-error-probability of the code in Example 1 with a 4-state encoder at the first level for the AWGN channel

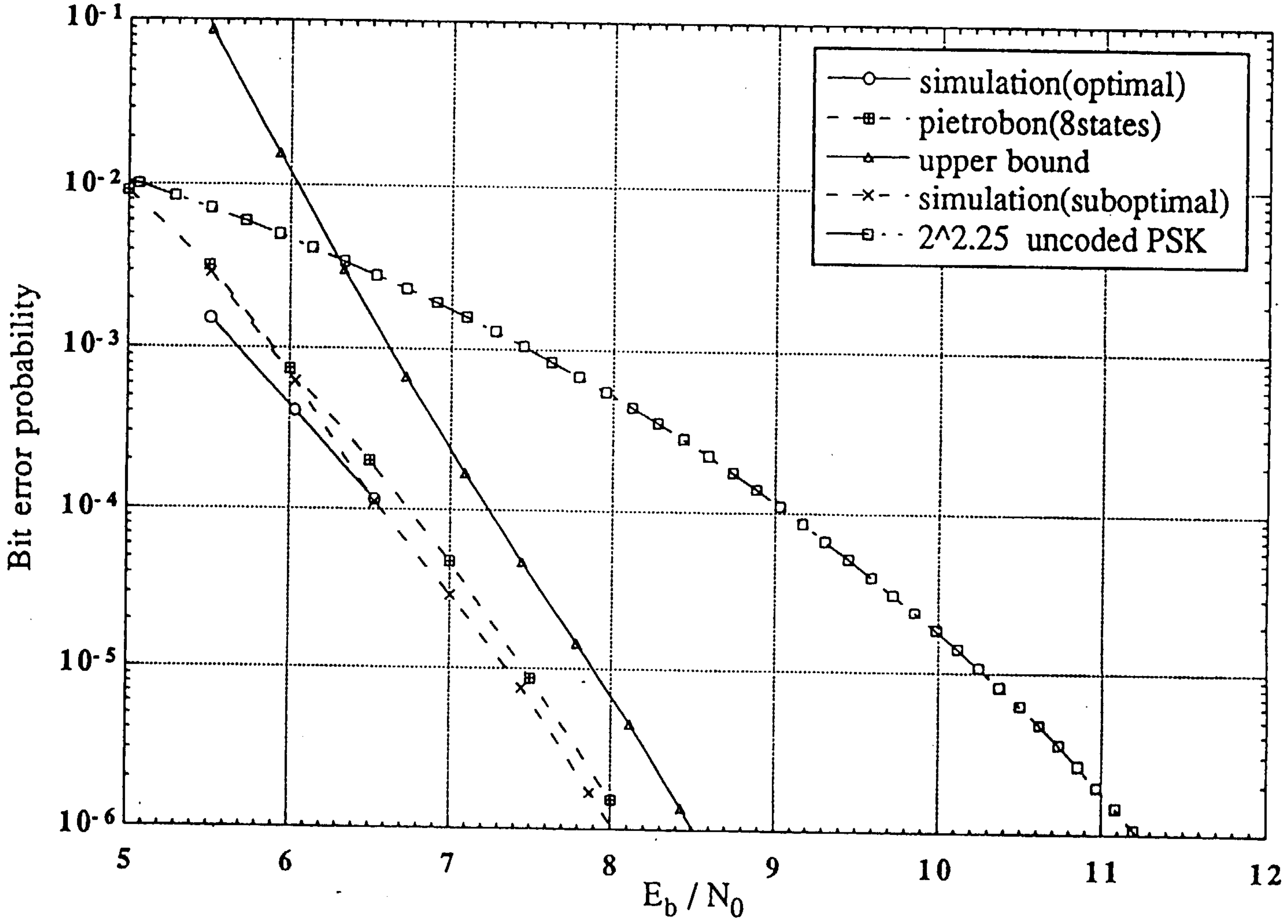

Figure 3 Bit-error-probability of the code in Example 1 with a 16-state encoder at the first level for the AWGN channel

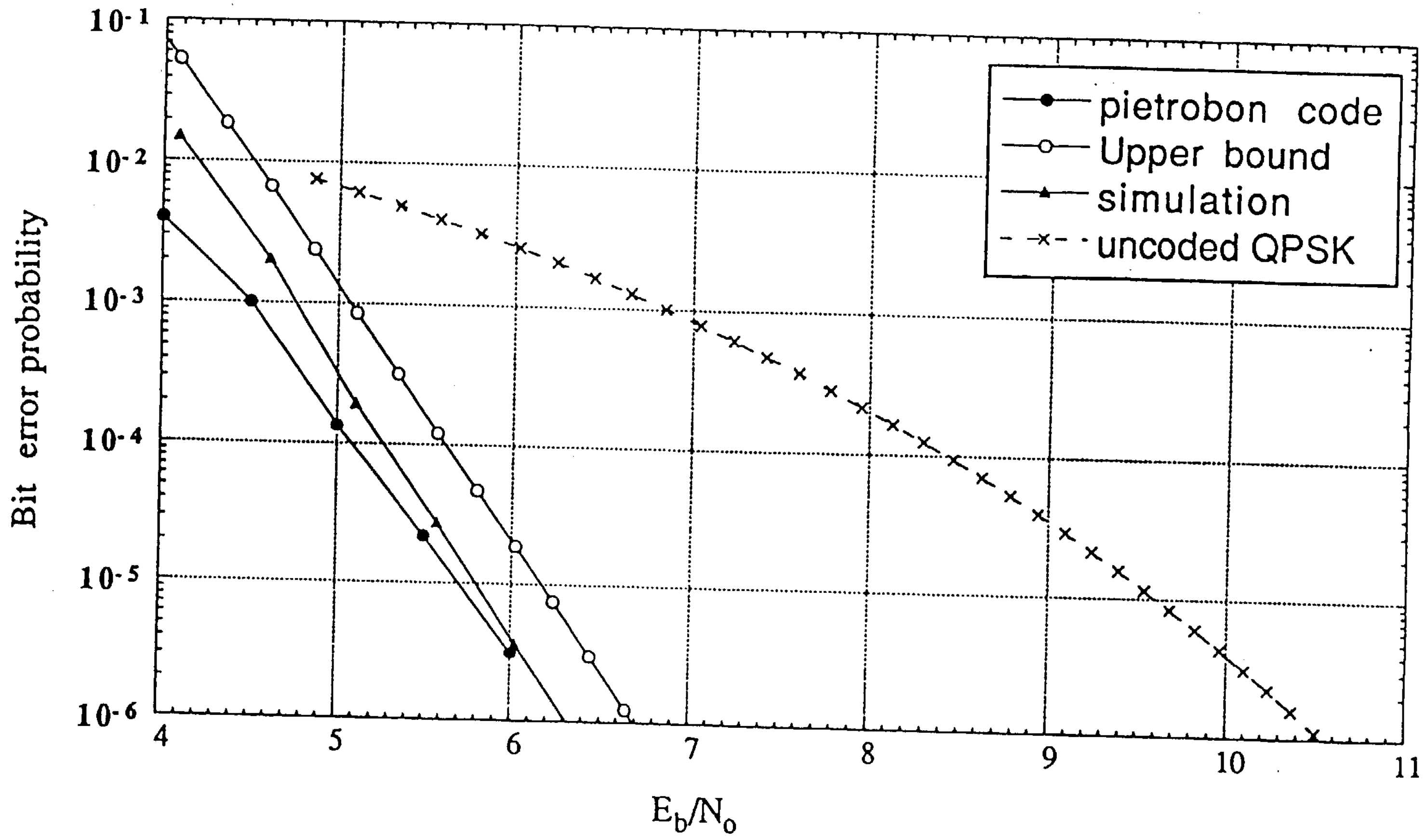

Figure 4 Bit-error-probability of the code in Example 2 for the AWGN channel

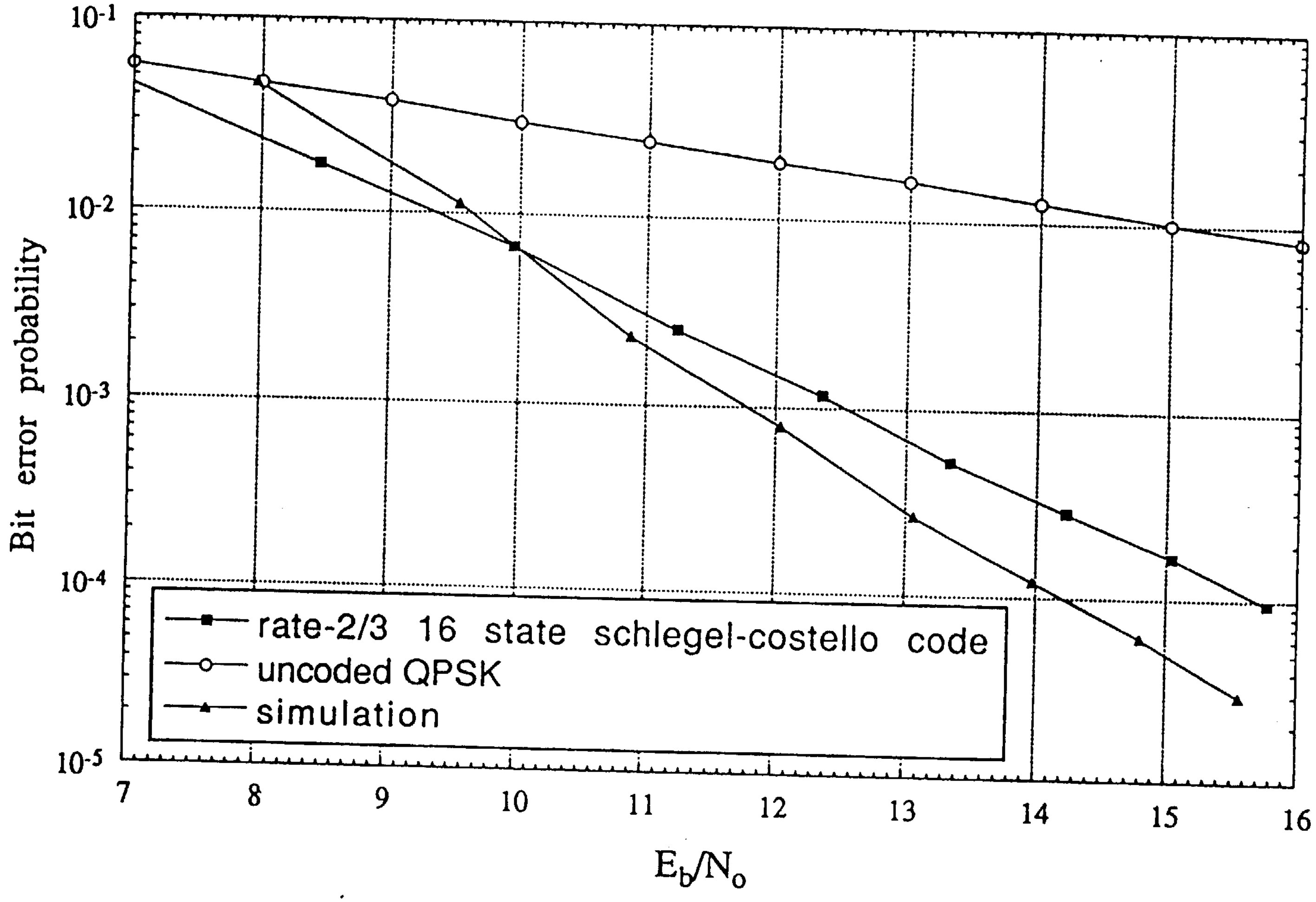

Figure 5 Bit-error-probability of the code in Example 3 for the Rayleigh fading channel

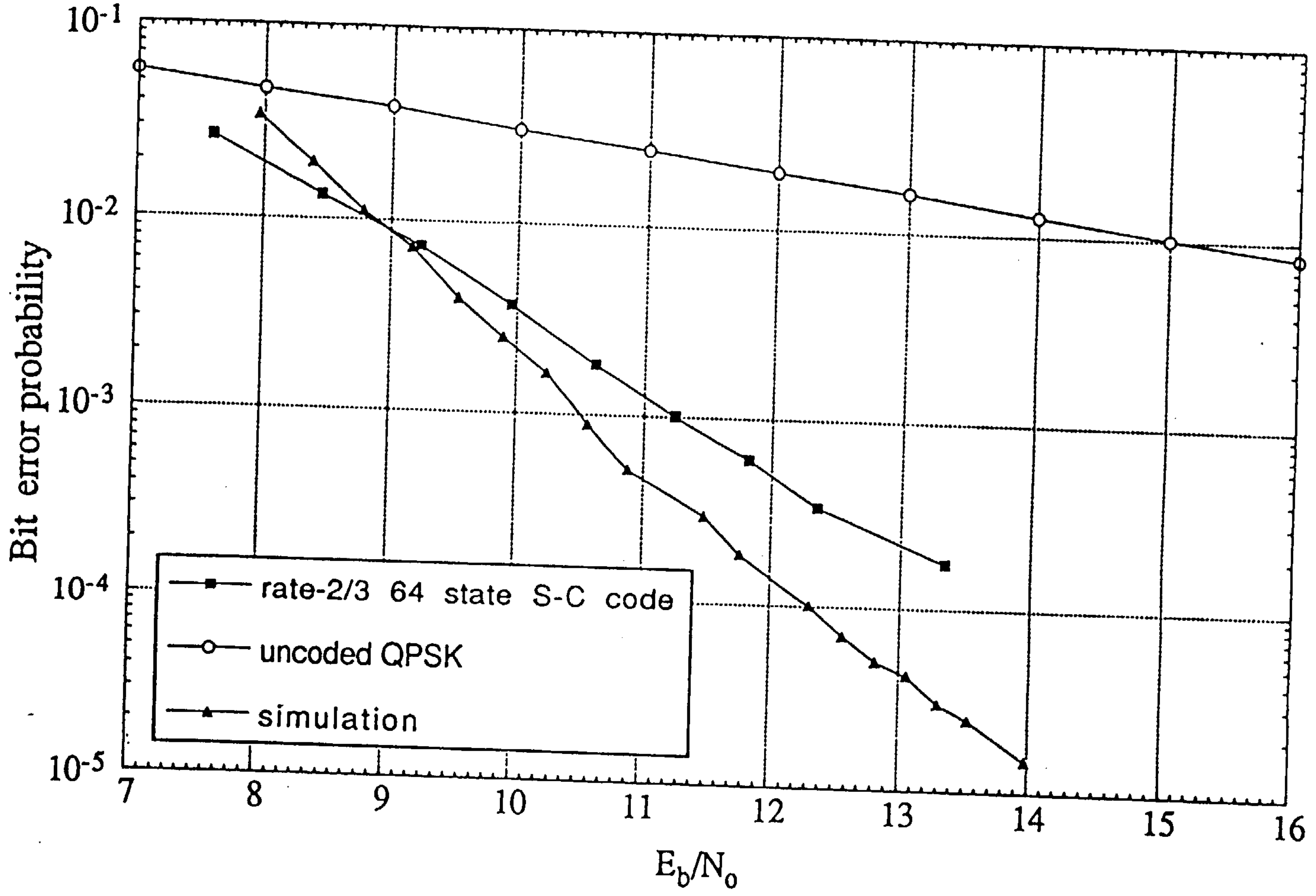

Figure 6 Bit-error-probability of the code in Example 4 for the Rayleigh fading channel

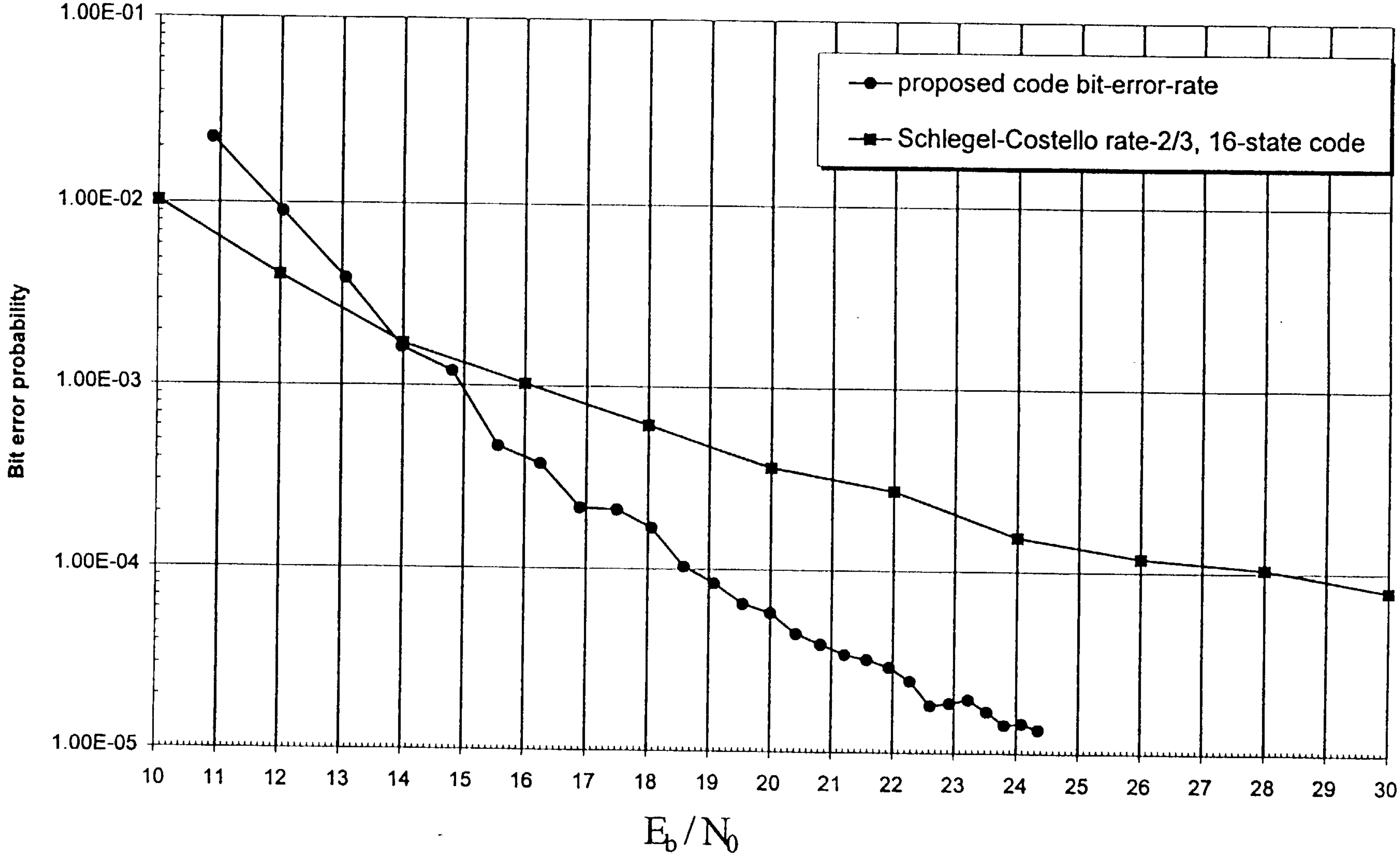

Figure 7 Bit-error-probability of the code in Example 5 for a shadowed mobile satellite channel